Herausgeber

M. HEIZMANN
T. LÄNGLE





M. Heizmann | T. Längle

FORUM BILDVERARBEITUNG 2020

FORUM BILDVERARBEITUNG 2020

Herausgegeben von
M. Heizmann und T. Längle



Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark
of Karlsruhe Institute of Technology.
Reprint using the book cover is not allowed.

www.ksp.kit.edu

Print on Demand 2020 – Gedruckt auf FSC-zertifiziertem Papier

ISBN 978-3-7315-1053-6
DOI 10.5445/KSP/1000124383

Vorwort

Bildverarbeitung spielt in vielen Bereichen der Technik und des Alltags eine wichtige Rolle zur Informationserfassung. Sie ist eine etablierte Technologie in der Mess- und Automatisierungstechnik. Wichtige Vorteile von Bildverarbeitungssystemen gegenüber anderen sensorischen Prinzipien bestehen u. a. darin, dass Bilder berührungslos gewonnen werden können und dass Bildsensoren inzwischen vergleichsweise günstig sind. Besonders spannend an der Bildverarbeitung ist aber, dass die Sensorik weitgehend dem menschlichen Leitsinn – dem Sehen – entspricht, aber nicht an die Beschränkungen des Menschen gebunden ist. Dies betrifft etwa die nutzbaren Spektralbereiche, die quantitative Interpretierbarkeit der Bilder, die gleichbleibende Aufmerksamkeit und Reproduzierbarkeit oder die Möglichkeit zur Erfassung hochdynamischer Prozesse. Auch wenn die Bildverarbeitung als Teil mehrerer Fachdisziplinen – u. a. Mess- und Automatisierungstechnik, Sytemtheorie, Mathematik, Informatik, Optik, Lichttechnik, Mikrosystemtechnik – eine gewisse Reife erreicht hat, ist dennoch von einer Sättigung von Forschung und Entwicklung keine Spur zu sehen. Ganz im Gegenteil: Gerade die Bildverarbeitung profitiert enorm von neuen Technologien wie z. B. dem maschinellen Lernen oder neuen Fertigungsmöglichkeiten für Mikrosysteme.

Das „Forum Bildverarbeitung“ hat das Ziel, über solche aktuellen Trends und neuartige Lösungen in der Bildverarbeitung zu berichten und zum fachlichen Austausch zwischen Wissenschaft und Anwendung beizutragen. Es findet in jedem zweiten Jahr seit 2010 statt und wird inzwischen gemeinsam vom Geschäftsfeld Inspektion und Optronische Systeme des Fraunhofer-Instituts für Optronik, Systemtechnik und Bildverarbeitung IOSB und dem Institut für Industrielle Informationstechnik IIIT des Karlsruher Instituts für Technologie KIT organisiert. Dem Aufruf zur Einreichung von Beiträgen sind erfreulich viele Autoren gefolgt. Aus den Einreichungen konnte der

Programmausschuss nach einer eingehenden Begutachtung über 35 hochwertige Beiträge auswählen und den Schwerpunkten

- Bildgewinnung,
- 3D-Verfahren,
- Bildverarbeitung,
- Maschinelles Lernen und
- Navigation

zuordnen. Zur überwiegenden Zahl der Beiträge wurden Aufsätze erstellt, die im vorliegenden Tagungsband enthalten sind.

Wir danken den Autoren für ihre Beiträge, den Mitgliedern des Programmausschusses für die Ansprache von Autoren und ihre wertvolle Expertise bei der Begutachtung der Einreichungen und allen, die durch ihre Anwesenheit zum Gelingen des Forums Bildverarbeitung beitragen. Für die Organisation der Veranstaltung und die technische Unterstützung bei der Erstellung des Tagungsbands bedanken wir uns bei Britta Ost und Jürgen Hock.

Auch das Forum Bildverarbeitung kommt an den Einschränkungen durch Corona nicht vorbei. Wir haben uns aber entschlossen, die Veranstaltung nicht ausfallen zu lassen und nach den rechtlichen Vorgaben bestmöglich für die Autoren und Teilnehmer durchzuführen. Wir hoffen, dass dieses Vorgehen für alle Beteiligten dennoch den erhofften fachlichen Austausch ermöglicht und bitten für die Einschränkungen um Verständnis.

Mitten in den Vorbereitungen zu diesem Forum Bildverarbeitung verstarb für uns unerwartet Fernando Puente León. Damit verlieren wir einen Fachkollegen, der das Forum Bildverarbeitung von Beginn an maßgeblich geprägt hat und dem das Ziel des fachlichen Austauschs in der Bildverarbeitung immer ein wichtiges Anliegen war. Wir führen das Forum Bildverarbeitung nun ohne ihn weiter und sind uns sicher, dass dies in seinem Sinne ist.

November 2020

Michael Heizmann
Thomas Längle

Wissenschaftliche Leitung

Prof. Dr.-Ing. M. Heizmann	Karlsruher Institut für Technologie
Prof. Dr.-Ing. T. Längle	Fraunhofer IOSB Karlsruhe

Programmausschuss

Prof. Dr. C. Bach	Buchs
Dr.-Ing. S. Bauer	Madison (Wisconsin)
Prof. Dr.-Ing. J. Beyerer	Karlsruhe
Prof. Dr. K. Donner	Passau
Dr. rer. nat. J. Eggert	Offenbach
Dr. T. Haist	Stuttgart
Prof. Dr. A. Heinrich	Aalen
Prof. Dr. B. Jähne	Heidelberg
Prof. Dr.-Ing. P. Lehmann	Kassel
Dipl.-Ing. M. Maurer	Wiesbaden
Prof. Dr. R. Neubecker	Darmstadt
Prof. Dr. F. Salazar	Madrid
Dipl.-Ing. M. Stelzl	Mainz
Prof. Dr. R. Stiefelhagen	Karlsruhe
Prof. Dr.-Ing. C. Stiller	Karlsruhe
Prof. Dr.-Ing. R. Tutsch	Braunschweig
Prof. Dr.-Ing. M. Ulrich	Karlsruhe
Prof. Dr.-Ing. S. Werling	Mannheim
Dr.-Ing. V. Willert	Darmstadt

Inhaltsverzeichnis

Bildverarbeitung

Maschinelles Lernen

Navigation

Characterization of Event-Based Image Sensors in Extent of the EMVA 1288 Standard

Alkhazur Manakov[1] and Bernd Jähne[2]

[1] Imago Technologies GmbH
Strassheimer Str. 45, 61169 Friedberg
[2] HCI at IWR, Heidelberg University,
Berliner Straße 43, 69120 Heidelberg

Abstract Recent years have seen a steady trend towards faster image sensors with higher resolution. It is well known that images and to a larger extent image sequence contain a lot of redundant information. An areas-scan image sensor, which is not sampled with a constant pixel and frame rate, but which outputs information only if something happens is therefore an interesting alternative. Such sensors are known as event-based or neuromorphic image sensors. Currently, there are several types of event-based image sensors on the market, but no universal concepts available to characterize these image sensors. In this work, we propose the characterisation concepts for neuromorphic sensors in extent of the EMVA standard 1288.

Keywords Sensor characterisation, event-based, neuromorphic

1 Introduction

In the recent years, state-of-the-art image sensors have seen a steady trend towards higher resolution and speed. The trend is driven by the need for faster and higher resolution vision systems in automotive, industrial and other fields. Despite of a significant progress made in the last decades, modern artificial vision systems are still much less effective and robust in solving real-world tasks than their biological counterparts. Even small insects outperform the most

powerful vision systems in such routine tasks as, for instance, real-time perception.

One of the limitations of the human-engineered vision systems is imposed by the image sensors and their principle of operation. Conventional sensors acquire the visual data in form of a series of images, recorded at discrete points of time. Visual data is sampled at a predetermined temporal intervals (frame rate) without any relation to the dynamics of the scene. On top of that, every image contains the data of all the pixels independently from whether this information, or part of it, has been recorded in previous images. This inflates the data rate unnecessarily and fast changes might be missed.

The alternative is the biologically inspired sensors: the dynamic vision sensors that implement the event-driven, frame-free approach. They are often referred to as "event-based" sensors due to their principle of operation. This family of sensors capture and is driven by the transient events in the visual scene, unlike conventional image sensors, that work with artificially created timing and control signals [1]. The latter implies that the control over the acquisition is transferred to single pixel, that handles its own information individually. The output of this sensor is compressed at the sensor level, thus optimizing data transfer, storage, and processing.

Characterisation of the conventional image sensors is a well known problem. The concepts, methodology and techniques for these sensors have been analysed, structured and resulted into the EMVA 1288 characterization standard [2]. These concepts cannot be applied to the event-based sensors. Characterization of event-based sensor is of a great importance, since it provides the means to compare them between each-other, and, most importantly, to conventional image sensors. In this paper, we address the problem of application-oriented characterisation of event-based sensors, establishing a link to EMVA 1288 standard, proposing characterisation techniques and presenting the first results. Dynamic vision sensor shows no response to static images. Therefore, new characterization concepts and procedures need to be developed, which take into account temporal aspect and can be applied to this type of sensors. At the same time, we would like to keep the possibility to compare the performance of the conventional image sensors with and the event-based ones.

2 Related work

2.1 Neuromorphic sensors

Biological retinas have many desirable characteristics, which are lacking in conventional image sensors, thus inspiring and driving the design of neuromorpic vision devices. Many of these advantageous characteristics have been modeled and implemented on silicon. Early development of such devices originated from the biological sciences community. The main purpose of these chips was to provide means for demonstration of neurobiological models and theories. Real-world applications were never the main focus. Therefore, very few of the sensors have been used in practical applications. Circuit complexity, large silicon area, low fill factors, or high noise levels and other factors prevented realistic applications [3], [4]. Recently, the development of practical vision sensors based on biological principles gained an increasing amount of attention and effort.

There is a family of event-based sensors, that encode illuminance in the time domain, namely in the rate of spike “events”. The pixels of these devices do not autonomously react to visual events in the scene. Thus, despite of having some advantages against conventional sensors, they fail to achieve redundancy suppression or latency reduction [4]. Large fixed pattern noise, complexity of the digital frame grabber and the big advantage of brighter pixel over the darker ones in allocating the communication bus make ”octopus retina” devices [5] impractical for conventional imaging. The pixels of so called ”time-to-first spike” imager [6], [7], [8] generate only one spike per frame, static parts of the scene generate spikes at the same time saturating the readout bus.

In dynamic vision sensor pixels are autonomous in detecting light changes in the scene. The gain in terms of temporal resolution with respect to conventional image sensors is dramatic. In addition, such parameters like the dynamic range of the scene greatly profit from the this approach. This type of sensor is very well suited for many machine vision applications including high-speed motion detection and analysis, object tracking, and shape recognition.

The pixel model proposed by Lichtsteiner et al. [9] simulates simplified three-layer retina (Figure 2.1). The circuit consists of a photo-

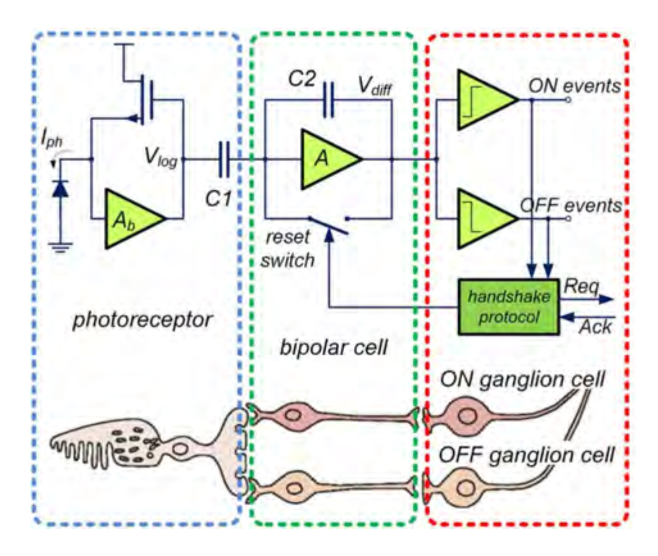


Figure 2.1: Simplified model of a human retina and corresponding event-based pixel circuitry. V_{log} tracking the photocurrent through the photo-receptor. The bipolar cell circuit responds with spike events V_{diff} of different polarity to positive and negative changes of the photocurrent. The ganglion cell circuit monitors the bipolar cells part and transports the spikes to the next processing stage.

receptor front-end, a differencing switched-capacitor amplifier and a comparator-based event generator. The photo-receptor responds logarithmically to irradiance, thus implementing a gain control mechanism that is sensitive to temporal contrast or relative change. The circuitry of the pixel allows to tune for the sensitivity of smaller or larger light changes in the scene. For instance, making the pixel biased to detect brighter-to-darker changes or vice versa. The parameters controlling the setting of the circuitry are called ”biases”. The pixels independently and asynchronously react to relative changes in intensity, producing sparse, frame-free, event-based output. Upon detection of the relative light intensity change the pixels communicate their state (ON or OFF) to the readout circuitry. The readout and the encoding circuitry encode the coordinated of the pixel, the state and the microsecond resolution time-stamp into an event-

packet. These packets or events can be gathered and analyzed by a visual inspection application.

The relative change events and gray-level image frames are two orthogonal representations of a visual scene. An event carries information about local relative changes, hence encodes all dynamic contents, yet there is no static parts of the scene. The conventional frame-based image is an absolute intensity map at a given point in time. Dynamic information is carried in form of blur. In principle, it is impossible to recreate change events from image frame nor can gray-level images be recreated from the events.

The most recent developments of sensor designs allow to combine the acquisition of static and dynamic information of the scene. Asynchronous time-based image sensor [1], [10] features fully autonomous pixels, that combine a change detector and a conditional exposure measurement circuit. The exposure measurement is initiated when an event is detected. Namely, the measurement starts immediately after the irradiance change of a certain magnitude has been detected by the respective pixel. Another recent approach to combine dynamic and static information into a single pixel is the so-called dynamic and active pixel vision sensor [11]. This pixel combines conventional frame-based sampling of intensity with asynchronous detection of log intensity changes. The advantages of combining the traditional and event-based vision comes at the cost of the capturing redundant output.

2.2 Conventional sensor characterization

Characterization of the conventional image sensors is a well known procedure. There are a number of techniques proposed for characterizing the property of certain sensor. The EMVA standard 1288 measures the mean (μ_y) and variance (σ_y^2) of the digital output signal as a function of the the pixel exposure in photons from dark to saturation [12]. With these measurements and a linear camera model it is possible to determine the signal-to-noise ratio SNR as a function

of the exposure per pixel in photons μ_p, neglecting the quantization noise:

$$\text{SNR}(\mu_p) = \frac{\mu_y}{\sigma_y} = \frac{\eta\mu_p}{\sqrt{\sigma_d^2 + \eta\mu_p}}. \tag{2.1}$$

For a linear sensor, SNR depends on the quantum efficiency η and the temporal variance of the dark signal σ_d^2. For a non-linear sensor, the input SNR is the most important quality parameter. It can be computed from the measured output SNR and the slope of the characteristic curve [13]:

$$\text{SNR}_{\textbf{in}}(\mu_p) = \frac{\mu_p}{\sigma_p} = \frac{\mu_p}{\sigma_y}\frac{\partial\mu_y}{\partial\mu_p} = \frac{\mu_p}{\mu_y}\frac{\partial\mu_y}{\partial\mu_p}\,\text{SNR}_{\text{out}}. \tag{2.2}$$

3 Event sensor characterization

These procedures are not applicable for the event-based sensors, because the latter are insensitive to static irradiation. Posch et al. [10] have initially addressed the problem of event-based sensor characterization. In their work, they have proposed a test method that allows simultaneously evaluating the main performance parameters and check how well the predictions from theoretical considerations agree with the performance of the sensor. We adapt the ideas proposed by Posch at al. [1] for the application-oriented characterization of the event-based sensors, in context of the EMVA 1288 characterization standard , described in the last section.

3.1 Properties

Sensitivity to small temporal contrasts, the response relation to the ON/OFF-biases settings and its uniformity across the array are crucial performance parameters for the asynchronous, event-driven sensors. The minimum detectable temporal contrast or simply *noise equivalent contrast* is barely detectable by an event-based pixel step change of the irradiation level. Noise equivalent contrast sensitivity corresponds to the signal-to-noise ratio property of a conventional image sensor as described in Sect. 2.2.

The sensitivity to the event-based sensor to the contrast is controlled by the ON- and OFF-biases. The biases might be set higher for making the sensor insensitive to small temporal contrast in the scene. The relation between the biases and the contrast threshold might be non-linear, depending on the circuitry of the pixel.

In the event-based sensor, the pixels react autonomously and asynchronously to the light transients in the scene. Therefore, the important characteristic of the sensor is *response uniformity*. In other words, how a single-pixel performance translates to the behaviour of the whole array. Due to production imperfection and tolerances the photo-sensors, circuitry will inevitably have variations in how pixel react to the same stimulus. This property of the event-based sensor corresponds to well-known nonuniformity property of the conventional image sensors.

3.2 Measurement procedure

The simplest way of experimentally determining the irradiation contrast $\Delta\mu_p$ necessary for generating one event for given mean irradiance level μ_p and event threshold settings is gradually increase the stimulus step until an event is generated. The stimulus' amplitude must initially be below the response threshold. It should also be fast enough, namely to have the rise time exceeding the bandwidth of the circuit under test. The minimal found stimulus amplitude always results in an event response when applied. In an ideal noise-free world, this would be the case and this method would be applicable.

In the real world conditions, the very same pixel will react differently to the same stimulus. Therefore, Posch et al. [10] propose to operate with "event probability" instead. It is defined for a given as ratio between the number of event responses M and the number of applied stimuli N, while background irradiance level and response thresholds remain unchanged. Plotting the "event probability" vs. stimulus amplitude, in an ideal noise-free world, would result in a step function. In reality, such curve would have an "S"-shape. Fitting the "S"-curve with a Gaussian error function, with the mean at the $M/N = 50\%$ event probability point, indicates the location of the barely sensible contrast.

Thus the irradiation contrast $\Delta\mu_p$ that produces an event probability of 50% corresponds to a temporal standard deviation of one sigma. In this way the input SNR can be measured as the ratio of μ_p and $\Delta\mu_p$ directly as a function of the mean irradiation from dark to saturation of the sensor. A linear response or the measurement of a characteristic curve is not required. This procedure corresponds to the new upcoming release of the EMVA standard 1288 for cameras with a non-linear response (Release 4.0 General, see [13]). The measurements are to be conducted on the entire array (or a selected area-of-interest) to ensure statistically significant conclusions, in the same way as for conventional cameras with the EMVA standard 1288.

Measurement procedure:

1. Choose ranges for background irradiance $[\mu_p; \mu_p^{max}]$ bias levels for ON/OFF events, stimulus amplitude $[\Delta\mu_p; \Delta\mu_p^{max}]$ and respective increments sizes.

2. For every background irradiance level, bias pairs and stimulus in the chosen range perform steps 3 and 4.

3. Apply stimulus and reset the selected pixels N times.

4. Count event responses M compute per-pixel probability P and for every of the selected pixels.

The data acquired this way from the whole sensor or part of it is sufficient to recover the contrast sensitivity, the response uniformity and the contrast threshold dependence on the bias settings.

4 First results

The experimental setup used for the measurements consists of an integrating sphere, background irradiance LED-lamp, contrast generation LED-lamp, the camera under test and the control PC. The data has been acquired and processed according to the procedure described above. The Figure 4.1 presents the event probability P dependence on the stimulus $(\Delta\mu_p/\mu_p)$ for different background irradiation levels. The data is acquired on the area of 128x128 pixels with

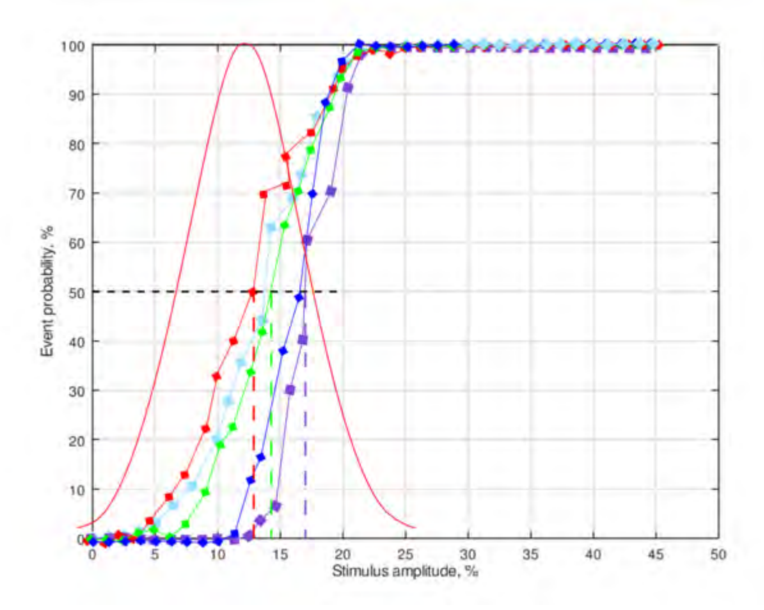


Figure 4.1: Event probability depending on the stimulus amplitude measured at one pixel. The ”S”-curves are acquired for different background irradiation levels. Higher irradiance levels correspond to steeper curves. Gaussian fit to the red ”S”-curve. The black dashed line indicates the even probability point. The vertical dashed lines indicate contrast thresholds for the corresponding curves.

the biases set 100 milliVolts. All the experiments were performed with VisionCam EB featuring Prophesee PPS3MVCD sensor.

The mean point of the Gaussian (50% event probability point) indicated ideal minimum contrast for event generation at this light level and for the chosen bias settings (Figure 4.1). In conventional sensors, this corresponds to a irradiation change of σ_p (Section 2.2). This means that the proposed method is able to measure the input SNR as a function of the irradiation. The response relation to the ON/OFF-biases settings can be extracted from the family of ”S”-curves. The contrast threshold grows with the background irradiance levels as represented by in Figure 4.1. The standard deviation

of the fitted Gaussian corresponds to the root-mean-square noise of the pixel.

5 Conclusions

In this work, we have adapted the concepts and methods developed by Posch et al. [10] to the application-oriented characterization of the event-based sensors, in terms of the EMVA 1288 characterizations standard. We have established the link between the properties of conventional and event-based sensors. Preliminary non-calibrated test measurements show that the measurement of the event probability is the correct way to measure the temporal noise of an event-based sensor and that this can be used to measured the SNR as a function of the irradiation. In this way event-based and conventional sensors can be compared directly. The analysis of the nonuniformities of event-based sensors requires further research.

References

1. C. Posch, D. Matolin, and R. Wohlgenannt, "A qvga 143 db dynamic range asynchronous address-event pwm dynamic image sensor with lossless pixel-level video compression," in *IEEE Conference of Solid-State Circuits*, 2010, pp. 400–401.
2. B. Jähne, "EMVA 1288 standard for machine vision – objective specification of vital camera data," *Optik & Photonik*, vol. 5, pp. 53–54, 2010.
3. S. C. Liu and T. Delbruck, "Neuromorphic sensory systems," *Current Opinion Neurobiology*, vol. 20, pp. 288–295, 2010.
4. D. Chen, D. Matolin, A. Bermark, and C. Posch, "Pulse modulation imaging – review and performance analysis," *IEEE Trans. Biomedical Circuits Systems in Cognitive Sciences*, vol. 5, no. 1, pp. 64–82, 2011.
5. E. Culurciello, R. Etienne-Cummings, and K. Boahen, "A biomorphic digital image sensor," *IEEE Journal Solid State Circuits*, vol. 38, no. 2, pp. 281–294, 2011.
6. Q. Luo and J. Harris, "A time-based cmos image sensor," in *IEEE International Symposium Circuits Systems*, vol. 4, 2004, pp. 840–843.

7. X. Qi, X. Guo, and J. Harris, “A time-to-first spike cmos imager,” in *IEEE International Symposium Circuits Systems*, vol. 4, 2004.
8. C. Shoushun and A. Bermak, “Arbitrated time-to-first spike cmos image sensor with on-chip histogram equalization,” vol. 15, no. 3, pp. 346–357, 2007.
9. P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change,” in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, 2006, pp. 2060–2069.
10. C. Posch and D. Matolin, “Sensitivity and uniformity of a 0.18µm cmos temporal contrast pixel array,” in *Circuits and Systems, ISCAS 2011, IEEE International Symposium*, 2011, pp. 1572–1575.
11. R. Berner, C. Brandli, M. Yang, S.-C. Liu, and T. Delbruck, “A 240 x 180 10 mw 12 us latency sparse-output vision sensor for mobile applications,” in *Symposium on Very Large Scale Integrated Systems*, Kyoto, Japan, 2013, pp. C186–C187.
12. EMVA 1288 Working Group, “EMVA Standard 1288 - standard for characterization of image sensors and cameras, release 3.1,” European Machine Vision Association, open standard, 2016.
13. B. Jähne, “Release 4 of the EMVA 1288 standard: Adaption and extension to modern image sensors,” *this volume*, 2020.

Release 4 of the EMVA 1288 Standard: Adaption and Extension to Modern Image Sensors

Bernd Jähne

Heidelberg University, HCI at IWR
Berliner Straße 43, 69120 Heidelberg

Abstract The well established and worldwide used EMVA Standard 1288 for objective camera characterization is still limited to linear monochrome or color cameras without preprocessing. This paper previews the upcoming Release 4.0 which can characterize a much wider range of imaging sensors. This includes sensors with an extended spectral range — especially into the short-wave infrared (SWIR) —, multispectral sensors with more than three color channels, polarization sensors, time-of-flight sensors, high-dynamic range image sensors and any other sensor with a non-linear characteristic curve, and sensors with preprocessing in the camera in order to optimize image quality.

Keywords Image sensor, cameras, standards, EMVA 1288

1 Introduction

The standard 1288 of the European Machine Vision Association (EMVA) is used worldwide for objective characterization of the quality parameters for industrial cameras [1–5]. It is the oldest standard activity of the EMVA. The standard has been elaborated by a consortium of the industry leading sensor and camera manufacturers, distributors, and research institutes. Work on the 1288 standard started in February 2004. A first version was published in 2005 [6] and the current release 3.1 went into effect end of 2016 [7]. This release can only be applied to cameras with a linear characteristic

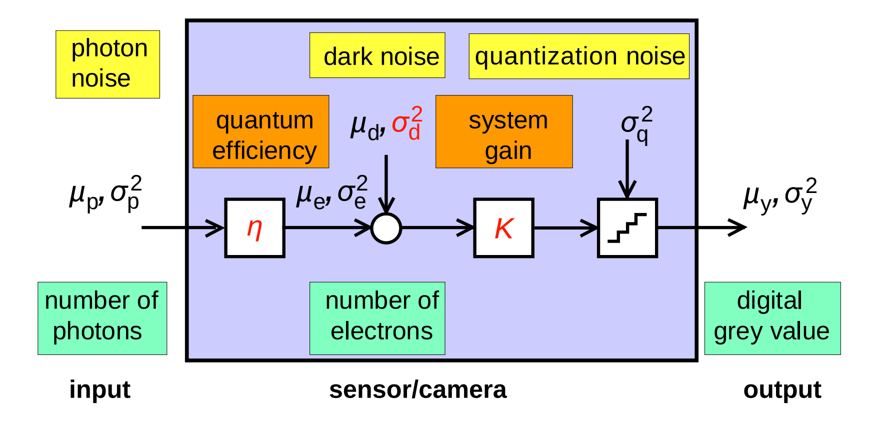


Figure 2.1: Linear model of a camera according to the EMVA 1288 standard.

curve. Furthermore, no preprocessing was possible which changes the temporal noise, except for simple operations such as binning or time-delayed-integration (TDI).

2 Linear model

The standard 1288 is generally based on a system theoretical concept which requires no measurements from within a camera. It is sufficient to measure the input signal, the mean number of photons μ_p hitting each pixel during the exposure time with a variance $\sigma_p^2 = \mu_p$ (Poisson process), and the output signal, the digital signal y (units DN) with mean μ_y and variance σ_y^2. No other measurements are required. With the current release 3.1 [7] a linear camera model is used with three unknown parameters (Fig. 2.1): the variance of the temporal dark noise σ_d^2 — subsuming *all* noise sources within the camera except for the quantization noise σ_q^2 —, the quantum efficiency η and the system gain K.

These three parameters can be determined from an irradiation series covering the whole range from dark to saturation measuring the

linear characteristic curve and the linear photon transfer curve (temporal noise variance versus mean of the digital camera signal) [7,8]:

$$\begin{aligned} &\text{Characteristic curve:} \quad \mu_y = \mu_{y.\text{dark}} \quad + K\eta\, \mu_p \\ &\text{Photon transfer curve:}\ \sigma_y^2 = K^2\sigma_d^2 + \sigma_q^2 + K(\mu_y - \mu_{y.\text{dark}}) \end{aligned} \tag{2.1}$$

The most important quality parameter of any measuring system it the signal-to-noise ratio SNR. From this quantity most application-oriented camera parameters such as the absolute sensitivity threshold, the dynamic range, and the maximum SNR can be derived [7]. For a linear system the input and output SNR are equal and can be computed from (2.1) resulting in

$$\text{SNR}(\mu_p) = \frac{\mu_y}{\sigma_y} = \frac{\eta\mu_p}{\sqrt{\sigma_d^2 + \sigma_q^2/K^2 + \eta\mu_p}} \tag{2.2}$$

Except for the influence of the quantization noise and provided that the temporal dark noise σ_d^2 does not depend on the system K, the SNR is — as expected for a linear system — independent of the system gain K.

The definition (2.2) does not yet include the signal degradation by spatial variations from pixel to pixel, which can also be described by a variance. For a linear system each pixel can have a different offset (dark signal nonuniformity DSNU) and slope (photo response nonuniformity PRNU). Therefore the spatial variance s_y^2 in units e^- can be expressed by

$$s_y^2 = \text{DSNU}_{1288}^2 + \text{PRNU}_{1288}^2 \left(\eta\mu_p\right)^2 \tag{2.3}$$

This spatial variances can be added to the temporal variances resulting in the total SNR

$$\text{SNR}_{\text{total}}(\mu_p) = \frac{\eta\mu_p}{\sqrt{\sigma_d^2 + \text{DSNU}_{1288}^2 + \sigma_q^2/K^2 + \eta\mu_p + \text{PRNU}_{1288}^2 \left(\eta\mu_p\right)^2}}. \tag{2.4}$$

An interesting aspect of this approach is that the performance of a real sensor can directly be compared with an ideal (the best possible)

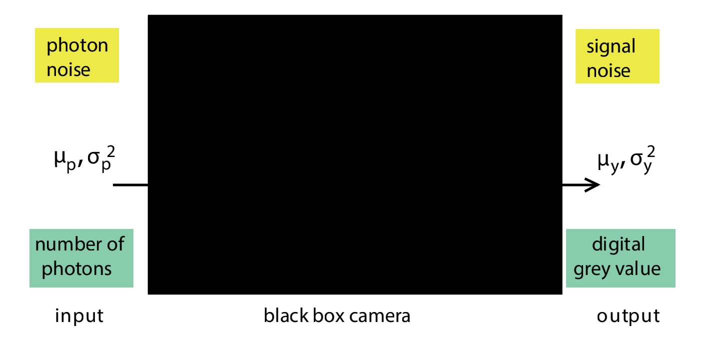


Figure 3.1: General black-box model of a camera according to the EMVA 1288 standard.

sensor. For an ideal sensor, the temporal dark noise, the quantization noise, DSNU and PRNU are zero and the quantum efficiency is one. Then (2.4) reduces to

$$\mathrm{SNR}_{\mathrm{ideal}}(\mu_p) = \sqrt{\mu_p} \tag{2.5}$$

3 General black-box model

For a camera with a non-linear characteristic curve, the linear model of EMVA 1288 release 3.1 cannot be applied. However, a camera with an arbitrary non-linear characteristic curve or a camera with preprocessing modifying the noise characteristics can be characterized by a true black-box model without *any* assumptions (Fig. 3.1). Even with this relaxed assumptions, the output SNR can be computed directly from the mean digital output signal and its temporal variance. It is also still possible to measure the characteristic curve $\mu_y(\mu_p)$ because it is the direct relation between the mean input and output signals.

For a general system the input SNR is different from the output SNR. The input SNR is the really important parameter for an image sensor. It gives the certainty with which the pixel irradiance can be measured. It is possible to compute the input SNR from the output

SNR because these two quantities are related to each other by the slope of the characteristic curve (laws of error propagation, [8]):

$$\mathrm{SNR_{in}} = \frac{\mu_p}{\sigma_p} = \frac{\mu_p}{\sigma_y}\frac{\partial \mu_y}{\partial \mu_p} = \frac{\mu_p}{\mu_y}\frac{\partial \mu_y}{\partial \mu_p}\mathrm{SNR_{out}} \quad (3.1)$$

It is important to note that the standard deviation σ_p does not only include the temporal noise of the incoming stream of photons (shot noise) but also all other noise sources within the non-linear camera — back-projected to the input signal.

It is also easy to specify the input SNR for an ideal general image sensor. Then there are no other noise sources and only the photon noise remains. Therefore the ideal input SNR is given — as for a linear camera (2.5) — by

$$\mathrm{SNR_{in.ideal}}(\mu_p) = \sqrt{\mu_p}. \quad (3.2)$$

In this way, it is possible to specify how much worse a real camera (3.1) is in comparison with an ideal one (3.2) also in the case of a general true black-box model. Without a more detailed camera model, it is not possible to determine the quantum efficiency[1] of the sensor. However, this is not a significant disadvantage. As with a linear camera (Sect. 2) the camera performance parameters really of importance for applications such as the absolute sensitivity threshold, the dynamic range, and the maximum SNR can be derived from the input SNR *without* knowing the quantum efficiency.

4 Fast and more detailed nonuniformity characterization

In order to analyze the spatial patterns of nonuniformities by the rich set of tools from the EMVA 1288 standard such as profiles, histograms and spectrograms [7], it is required to suppress the temporal noise. Because the spatial and temporal variances are roughly of the same order of magnitude, this requires averaging over hundreds of images. This is no real problem for a linear camera because it is

[1] The quantum efficiency relative to a maximum response can still be measured by performing measurements over the whole range of wavelengths.

sufficient to analyze the nonuniformities with just two parameters, the DSNU and PRNU (Sect. 2). Thus averaging over many images is only required for the dark images and images at 50% saturation [7].

With the general black-box model (Sect. 3), averaging at just two irradiation levels is generally not sufficient. The best would be to estimate the spatial nonuniformity at all irradiation levels where the temporal noise is measured. These are at least 50 levels [7]. Therefore this approach is not feasible. Thus the question arises whether it is possible to determine at least some significant parameters of the spatial nonuniformity with much fewer images.

4.1 Temporal and spatial variances from just two images

In the following, a new approach is detailed which works with much fewer images — as few as two images are sufficient — and still provides a detailed statistical analysis of the spatial nonunifomities.

The starting point is the observation that the stationary nonuniformities can entirely be eliminated by computing the temporal noise from the difference of two images taken with the same irradiation. This is the approach taken in the EMVA standard 1288 to compute the variance of the temporal noise [7]. The mean from two images $\boldsymbol{y}[0]$ and $\boldsymbol{y}[1]$ is

$$\mu = \frac{1}{2NM} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (y[0][m][n] + y[1][m][n]) \tag{4.1}$$

and the temporal variance computed from the difference image is

$$\sigma^2 = \frac{1}{2NM} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (y[0][m][n] - y[1][m][n])^2. \tag{4.2}$$

The variance computed in this way must be divided by two because the variance of the difference image is two times higher than the variance of a single image.

The key point is now that single images contains both the temporal noise and the spatial nonuniformity. In this way, the spatial nonuniformity can be computed by subtraction. However, it must be ensured that the subtraction never results in negative variances. In the following, it is shown under which conditions this is possible.

In order to simply the equations, the following abbreviations are introduced for the mean value and the variance of image $\boldsymbol{y}[l]$[2]

$$\begin{aligned}
\mu[l] &= \frac{1}{NM}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} y[l][m][n] = \overline{\boldsymbol{y}[l]} \\
\sigma^2[l] &= \frac{1}{NM}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} (y[l][m][n]-\mu_y[l])^2 = \overline{(\boldsymbol{y}[l]-\mu_l)^2}
\end{aligned} \tag{4.3}$$

In the ideal case mean values of all images at the same irradiation level have the same mean value, but they may be different and it is required to check whether this effect can result in negative variances.

The image is assumed to be composed of a mean value $\mu[l]$, differing from image to image, a zero-mean temporal noise signal $\boldsymbol{n}[l]$ with a variance σ^2 and a zero-mean and stationary spatially varying signal $\boldsymbol{s}$ with a variance s^2:

$$\boldsymbol{y}[l] = \mu_l + \boldsymbol{n}[l] + \boldsymbol{s}, \quad \overline{\boldsymbol{y}[l]} = \mu_l, \quad \overline{(\boldsymbol{y}[l]-\mu_l)^2} = \sigma^2 + s^2. \tag{4.4}$$

Furthermore it is assumed that the temporal noise from different images and the temporal noise and spatial nonuniformities are statistically independent, i. e. $\overline{\boldsymbol{n}[l]\boldsymbol{n}[k]} = 0$ (if $k \neq l$) and $\overline{\boldsymbol{n}[l]\boldsymbol{s}} = 0$.

Now two terms are evaluated. Firstly, the temporal variance from the difference image (4.2) needs to be corrected for possible different mean values of the two images

$$\begin{aligned}
A &= \overline{[(\boldsymbol{y}[0]-\mu_0)-(\boldsymbol{y}[1]-\mu_1)]^2} \\
&= \overline{\boldsymbol{y}^2[0]} + \overline{\boldsymbol{y}^2[1]} - 2\overline{\boldsymbol{y}[0]\boldsymbol{y}[1]} - (\mu_0-\mu_1)^2 \stackrel{!}{=} 2\sigma_y^2
\end{aligned} \tag{4.5}$$

Secondly, the variances of the two images are added up, which include both the variances of the temporal noise and the spatial nonuniformity:

$$\begin{aligned}
B &= \overline{[(\boldsymbol{y}[0]-\mu_0)^2} + \overline{[(\boldsymbol{y}[1]-\mu_1)^2} \\
&= \overline{\boldsymbol{y}^2[0]} + \overline{\boldsymbol{y}^2[1]} - \mu_0^2 - \mu_1^2 \stackrel{!}{=} 2\sigma^2 + 2s^2
\end{aligned} \tag{4.6}$$

[2] Both sums are divided by the total number of pixels NM, although for a bias-free estimate of the variance the divisor should be one less ($NM-1$). This approach is necessary to have the same averaging scheme for means and variances. The error introduced is very small and can even be corrected at the end by multipling the estimated variances with $NM/(NM-1)$.

The difference of the two terms should be equal to s^2 and therefore always be positive:

$$B - A = 2\overline{\boldsymbol{y}[0]\boldsymbol{y}[1]} - 2\mu_0\mu_1 \stackrel{!}{=} 2s^2. \tag{4.7}$$

This can be verified by inserting the image signal (4.4) into (4.7).

It is essential to include the possibly slightly different mean values of the two images in term A (4.5). If this is not done, the term $B - A$ could become negative and too high temporal variances and too low spatial variances are computed:

$$2\overline{\boldsymbol{y}[0]\boldsymbol{y}[1]} - \mu_0^2 - \mu_1^2 = 2s^2 - (\mu_0 - \mu_1)^2 \neq 2s^2. \tag{4.8}$$

4.2 Split into row, column, and pixel nonuniformities

Modern CMOS sensors may exhibit not only pixel-to-pixel nonuniformities, but also row-to-row and/or column-to-column nonuniformities. Therefore it is important to decompose the spatial variance into row, column, and pixel variances:

$$s^2 = s^2_{\mathrm{row}} + s^2_{\mathrm{col}} + s^2_{\mathrm{pixel}}. \tag{4.9}$$

All three unknowns can still be estimated by computing additional spatial variances from a rows and columns averaged over the whole image. The mean row and column of a single image are given by

$$\mu[n] = \frac{1}{M}\sum_{m=0}^{M-1} y[m][n], \quad \mu[m] = \frac{1}{N}\sum_{n=0}^{N-1} y[m][n]. \tag{4.10}$$

The column spatial variance computed from the average row

$$s^2_{\mathrm{col}} = \frac{1}{N-1}\sum_{n=0}^{N-1}(\mu[n] - \mu)^2 \quad - \quad s^2_{\mathrm{row}}/N - s^2_{\mathrm{pixel}}/N - \sigma^2/(N) \tag{4.11}$$

still contains a residual row spatial variance, pixel spatial variance and temporal variance. Averaging over N rows does not completely suppress theses variances. Therefore the three terms on the right

hand need to be subtracted. Likewise, the *row spatial variance* computed from the average column

$$s^2_{\text{row}} = \frac{1}{M-1}\sum_{m=0}^{M-1}(\mu[m]-\mu)^2 \quad - \quad s^2_{\text{col}}/M - s^2_{\text{pixel}}/M - \sigma^2/(M) \tag{4.12}$$

contains residual column spatial variance, pixel spatial variance and temporal variance.

The three equations (4.9), (4.11), and (4.12) form a linear equation system from which all three components of the spatial variance can be computed. With the two abbreviations

$$\begin{aligned} s^2_{\text{cav}} &= \frac{1}{N-1}\sum_{n=0}^{N-1}(\mu[n]-\mu)^2 \;\; - \sigma^2/(N), \\ s^2_{\text{rav}} &= \frac{1}{M-1}\sum_{m=0}^{M-1}(\mu[m]-\mu)^2 - \sigma^2/(M), \end{aligned} \tag{4.13}$$

the linear equation system reduces to

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1/M & 1/M \\ 1/N & 1 & 1/N \end{bmatrix} \begin{bmatrix} s^2_{\text{col}} \\ s^2_{\text{row}} \\ s^2_{\text{pixel}} \end{bmatrix} = \begin{bmatrix} s^2 \\ s^2_{\text{rav}} \\ s^2_{\text{cav}} \end{bmatrix}. \tag{4.14}$$

The solution of this linear equation systems is

$$\begin{aligned} s^2_{\text{col}} &= \frac{M}{M-1}s^2_{y.\text{rav}} - \frac{1}{M-1}s^2, \\ s^2_{\text{row}} &= \frac{N}{N-1}s^2_{y.\text{cav}} - \frac{1}{N-1}s^2, \\ s^2_{\text{pixel}} &= \frac{MN-1}{(M-1)(N-1)}s^2 - \frac{N}{N-1}s^2_{\text{cav}} - \frac{M}{M-1}s^2_{\text{rav}}. \end{aligned} \tag{4.15}$$

Negative variances cannot result from this split up of the variances, because they are calculated from single images. Therefore changes in the mean values from image to image do not influence the computation. However it is important to avoid numerical rounding errors by choosing an appropriate high-accuracy arithmetic and suitable algorithms.

4.3 Variances of temporal noise and nonuniformity; signal stability

The two new schemes detailed in the previous two sections enable the computation of the variances of both the temporal noise and the spatial nonuniformity from just two images at one irradiation level. It is even possible to split the latter into variations from row to row, column to column, and pixel to pixel.

The difference in the mean values between two images at the same irradiation level carries also an important additional information, namely how stable the irradiation measurement is from image to image.

For any camera the spatial nonuniformity is analyzed at all irradiation levels. For a camera with an arbitrary non-linear characteristic curve, then the most critical irradiation levels can be chosen from these measurements, where it is useful to average over hundreds of images to apply further tool of the EMVA standard 1288 such as profiles, histograms and spectrograms [7] for a more detailed analysis of the spatial patterns.

5 Further comprehensive extensions

In order to cope with modern image sensors release 4.0 includes many further extensions. In this section the most important of them are briefly described.

- The wavelength range is extended from deep UV to SWIR. In the deep UV, when more than one charge unit is produced by a single photon, the simple linear model can no longer be used even for a linear sensor, because a new noise source arises.
- With the general model also image intensifiers including emCCDs can be characterized.
- Raw data of *any* image modality can be characterized according to the standard
- Characterization of the polarisation angle and degree of polarization of a polarization image sensor is an example for the characterization of parameters derived from multiple channels.

The rich set of tool of the standard can also be applied to such parameters.

- Optionally, cameras with lenses or an illumination corresponding to a given exit pupil can be measured. In this way it is possible to measure also image sensors with micro lenses that are shifted towards the edge of the sensor.
- The new version includes a better measure for the linearity of the characteristic curve than in release 3.1. Because the slope of the characteristic curve is evaluated according to the general model (Sect. 3), also the differential non-linearity is known.

6 Conclusions

The new Release 4.0 adequately considers the rapid progress of imaging sensors. It will be possible to characterize a much wider spectrum of cameras/sensors: UV and SWIR-sensitive, multispectral, polarization, intensified (such as EM-CCDs), multilinear and highdynamic range. Also, cameras with lenses and preprocessing to enhance the image quality can be characterized. Despite the diversity, the quality of cameras can still be described with a minimum set of application-oriented quality parameters. It is planned to publish a release candidate of Release 4.0 in the fall of 2020.

The rich tool set of the EMVA 1288 standard to characterize temporal noise, nonuniformity and defect pixels can also be applied to any parameters derived from several channels of a multimodal image sensor. As a prime example the analysis of the degree of polarization and polarization angle computed from a polarization image sensors is contained.

The new release 4.0 does not yet cover an entirely different class of image sensors, so-called event-based or neuromorphic sensors. Research to extend the EMVA standard 1288 also for this class of sensors has already started [9].

7 Acknowledgments

The author gratefully acknowledges financial support for this research through his senior professorship, jointly funded by the Rector of Heidelberg University, HCI and IWR. The discussions within the EMVA 1288 working group were also very helpful in developing the new general model for cameras with an arbitrary characteristic curve and/or preprocessing.

References

1. A. Darmont, "Using the EMVA 1288 standard to select an image sensor or camera," in *Sensors, Cameras, and Systems for Industrial/Scientific Applications XI*, ser. Proc. SPIE, E. Bodegom and V. Nguyen, Eds., vol. 7536, 2010, p. 753609.
2. B. Jähne, "EMVA 1288 standard for machine vision – objective specification of vital camera data," *Optik & Photonik*, vol. 5, pp. 53–54, 2010.
3. A. Darmont, J. Chahiba, J. F. Lemaitre, M. Pirson, and D. Dethier, "Implementing and using the EMVA1288 standard," in *Sensors, Cameras, and Systems for Industrial/Scientific Applications XIII*, ser. Proc. SPIE, R. Widenhorn, V. Nguyen, and A. Dupret, Eds., vol. 8298, 2012, p. 82980H.
4. M. Rosenberger, C. Zhang, P. Votyakov, M. Preißler, R. Celestre, and G. Notni, "EMVA 1288 camera characterisation and the influences of radiometric camera characteristics on geometric measurements," *Acta IMEKO*, vol. 5, pp. 81–87, 2016.
5. A. Darmont, *High Dynamic Range Imaging: Sensors and Architectures*, 2nd ed. SPIE, 2019.
6. EMVA 1288 Working Group, "EMVA Standard 1288 - standard for characterization of image sensors and cameras, release A1.00," European Machine Vision Association, open standard, 2005.
7. ——, "EMVA Standard 1288 - standard for characterization of image sensors and cameras, release 3.1," European Machine Vision Association, open standard, 2016.
8. B. Jähne, *Digitale Bildverarbeitung und Bildgewinnung*, 7th ed. Berlin: Springer Vieweg, 2012.
9. A. Manakov and B. Jähne, "Characterization of event-based image sensors in extent of the EMVA 1288 standard," in *this volume*, 2020.

Light Field Illumination: A Universal Lighting Approach for Visual Inspection

Christian Kludt, Lukas Dippon, Thomas Längle, and Jürgen Beyerer

Fraunhofer IOSB,
Fraunhoferstr. 1, 76131 Karlsruhe

Abstract Choosing a proper lighting approach is a crucial task in designing visual inspection systems. It becomes especially challenging for complex-shaped objects, which change the direction and distribution of incoming light in various ways. We overcome this challenge by constructing a light field display and deploy it as a highly tunable lighting device. By programmatically controlling the spatial position of the individual light sources while simultaneously controlling their angular direction of emission, an object-specific light field can be generated, which highlights the features of the object under test with maximum contrast. We explain the calibration procedure, the rendering pipeline and present first results of the device's performance.

Keywords Light field, visual inspection, programmable lighting, projection calibration, phase-shifting, light field-rendering

1 Introduction

The image processing chain consists of the main components illumination, light-material interaction (transmission, deflection, reflection, scattering, etc.), image acquisition, digitization, data evaluation, classification and finally decision making. For the latter to be carried out robustly, it is crucial to extract the key features of the test object with high contrast. It is usually advantageous to control the image creation process at the beginning, i. e. the illumination. The most common illumination setups are shown in Figure 1.1: bright and dark

field illumination both with front and back lighting. Depending on the direction of the incoming light, different structures of the object are highlighted.

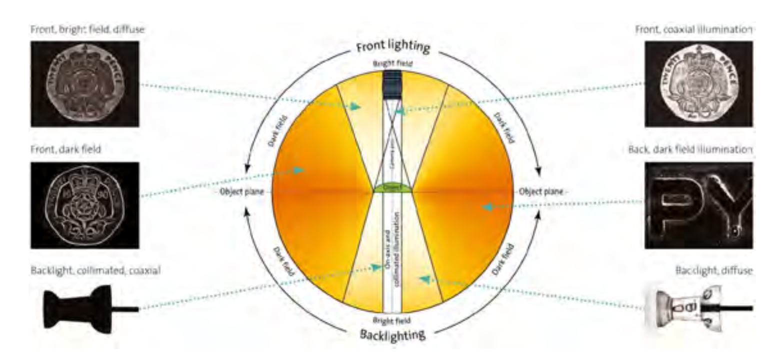


Figure 1.1: Illumination setups. From [1]. By varying the angle of illumination, different structures of the specimen become visible.

However, more complex objects require a more sophisticated illumination setup, which is realized by adding further light sources. Currently, we deploy machine vision systems with more than 64 different lighting channels. Adding, adjusting and testing them is a time consuming process, particularly during system design. Different lighting hardware such as collimated, diffuse, structured and colored lightings have to be evaluated. It would come in handy to have more generic lighting approach in a single device.

2 Existing Approaches

The Purity [2] inspection system at Fraunhofer IOSB utilizes a multichannel imaging system to acquire images from various illumination directions at once. The basic setup is depicted in Figure 2.1. For each kind of defect, a suitable illumination channel is realized which ensures an image with maximum contrast. Opaque inclusion will appear dark in the bright field channel (red), whereas scattering defects such as enclosed air bubbles will appear as bright spots in the dark field image (blue). Scattering defects on the surface, e. g. scratches or dust, become visible under gracing illumination (green).

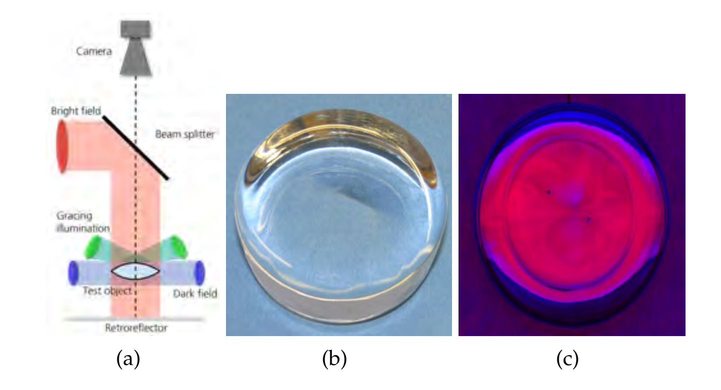


Figure 2.1: Purity inspection system. From [3]. (a) Basic setup. (b) Transparent test object. (c) Acquisition of different illumination directions simultaneously by means of color multiplexing.

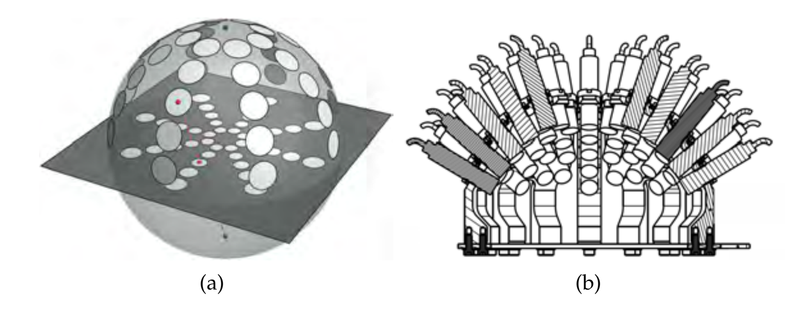


Figure 2.2: Hemispherical illumination. (a): Lighting pattern of Gruna [4]. (b) Setup of Schöch et al. [5].

Gruna [4] as well as Schöch et al. [5] both utilize several individual light sources, which are located on a hemisphere, cf. Figure 2.2. By activating them individually or in groups, the direction of illumination can be controlled in a targeted manner. However, the illumination direction of the individual elements cannot be altered once they are adjusted. Therefore, we propose a light field display as a universal lighting approach for visual inspection systems.

3 Proposed Method

A light field display is a planar light source in which both the position and the direction of light emission can be controlled independently and simultaneously. Our prototype combines a monitor with an array of lenses mounted in front of it at a distance corresponding to the focal length of the individual lenses, cf. Figure 3.1. Whenever a pixel is activated behind an individual lens, the light field display emits a parallel bundle of rays in a direction determined by the spatial position of the activated pixel behind the individual lens. This generates a 4D light field and enables a customizable illumination from many individual spatial positions and angular directions at once.

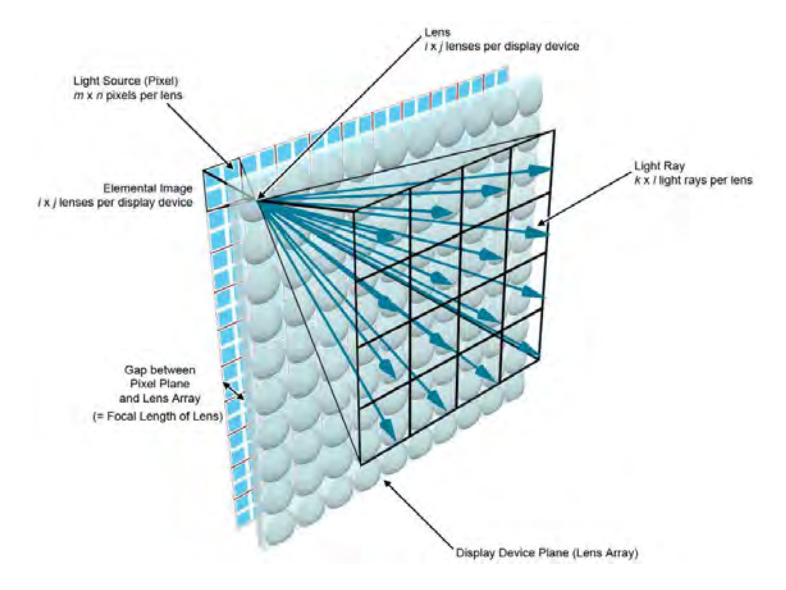


Figure 3.1: Basic setup of a light field display. From [6]. Both the spatial position and the angular direction of the light emission ca be controlled simultaneously.

4 Technical Realization

The light field emitted by a light field display can be adjusted in four dimensions. It allows for two degrees each of spacial and angular freedom. Since the light field display converts between the two-dimensional image displayed on its monitor and the emitted light field purely physically, the challenge in displaying the wanted light field lies with converting the four-dimensional input into a two-dimensional representation which can be displayed by the monitor.

$$L(x, y, \phi, \gamma) \xrightarrow{rendering} I(u, v) \xrightarrow[conversion]{optical} L'(x, y, \phi, \gamma) \quad (4.1)$$

The exact transformation required is determined by a variety of factors such as the pixel density of the screen used, the size and optical properties of each micro-lens and the arrangement of these lenses relative to the screen pixels.

In order to be able to control the light field, it is necessary to know precisely which monitor pixel is responsible for emitting light in a specific direction. The assignment of each monitor pixel to a spatial pixel (individual lens) and a directional pixel (observation angle) is conducted by means of the following calibration routine.

Calibration The goal is to determine the exact position of the central pixel behind each micro lens. Light emitted from these center pixels is collimated by the lens array into a bundle of rays that extends perpendicularly from the monitor.

The position of the center pixels corresponds to the position of the individual lenses in the plane of the lens array. Based on this relationship, the emission angles of the neighboring pixels can be computed.

To achieve this, we place a camera with a telecentric lens in front of the light field display with its optical axis aligned vertically to it. Now the captured light rays origin from these very center pixels.

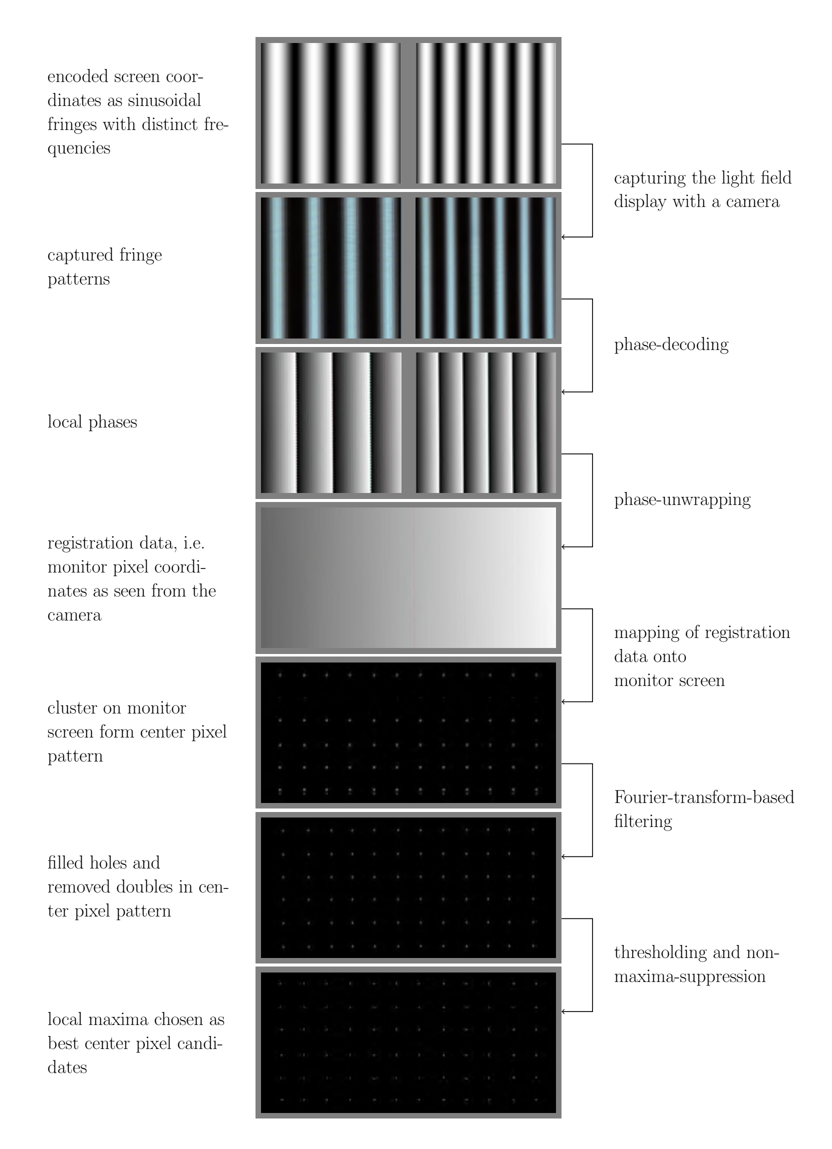


Figure 4.1: Visual representation of the calibration procedure. The goal is to determine the exact position of the center pixels behind each micro-lens.

We determine their exact positions by means of a coded illumination sequence. We deploy temporal phase shifting with two sets of sinusoidal fringes with different wavelength each. Once their phases are decoded locally, their global position is computed by multi frequency temporal phase unwrapping. Here, each camera pixel acts as an individual sensor as opposed to spatial unwrapping approaches, where neighboring pixels are used. Hence, the unwrapping is robust when measuring complex objects with discontinuities and isolated surfaces such as the one of the micro-lens array.

Mapping the number of occurrences of each position results in clusters around the center pixel positions. Their spread (Figure 4.2) origins mainly from the spherical shape of the individual lenses, not focusing in a point but rather in a point spread function. Also, whenever the optical axis of a micro-lens does not intersect with the center of a single pixel, but rather with the intersection of two (four) neighboring ones, the distribution will be spread at least over those two (four) pixels. Since the spatial distance between monitor pixels and micro-lenses are not integer multiples of each other, these two cases will alternate periodically. This effect is known as incommensurability. As a result, the light field display is unable to emit perfectly collimated light. However, the effect is rather small, cf. Figure 5.3.

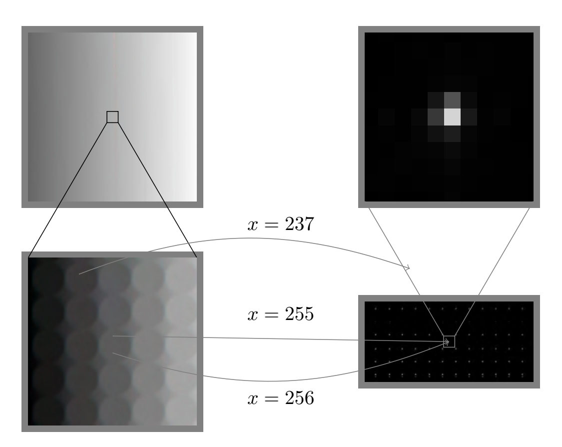


Figure 4.2: Mapping the registration data onto the monitor screen reveals clusters around the center pixel positions.

Error Correction Although the calibration process is very robust, disturbing factors can lead to errors in the decoded positions of center pixels. Most notably, the occlusion of lenses or parts thereof can result in center pixels whose position is not represented by occurrences above the general noise level. The position of these is therefore lost in the process. In order to avoid such false negatives, additional information in form of global regularity in the position of center pixels can be used. This requires the micro-lenses to actually be arranged in a regular grid, which is the normal case for implementations of light field displays. Hence, this regularity results in distinct maxima when transformed to the Fourier space. Any error constitutes a disturbance of the regular pattern, which in turn adds additional frequencies, cf. Figure 4.3. Their amplitudes correlate with the disturbance's fraction within the pattern. Since these errors generally do not occur following a regular pattern, and their occurrence is rather rare, the amplitudes of these frequencies are quite small and can be filtered by thresholding. After inverse Fourier transformation, this yields a map of center pixels with filled holes and removed doubles. Finally, we simply approximate each center pixel by the integer pixel closest to its optimal position.

Rendering The center pixels form the base for rendering 2D-representations of light fields. The spatial dependency of the light field is controlled by addressing all pixels behind one micro-lens, i. e. all pixels which are closest to one distinct center pixel.

The calibration procedure is not limited to determining the center pixels of each lens, but will more generally yield the positions of all pixels emitting light in any direction the camera captures them from. Therefore, by repeating the procedure with the camera turned by the horizontal and vertical angles ϕ and γ relative to the light field display's normal, one can obtain the positions of all pixels emitting light in this direction. As described by Meyer et al. [7], the resulting pattern is the same as that of the center pixels, but translated by a certain pixel offset Δu and Δv.

Alternatively, we can exploit the geometrical dependencies of the light field display. Assuming the pixel size m and the focal length f of the lenses is known and consistent throughout the micro-lens

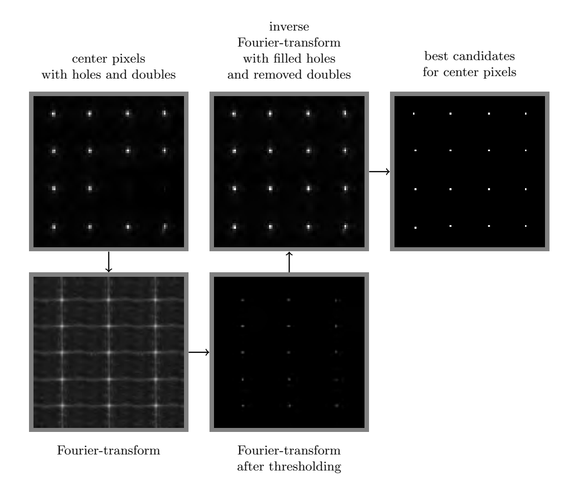


Figure 4.3: Error correction. Irregularities in the grid such as holes and doubles are compensated for using a threshold filter in Fourier domain.

array, determining the angle in which a pixel at a distance Δu resp. Δv relative to the center pixel of each lens will emit light is straight forward:

$$\phi = \arctan\left(\frac{m \cdot \Delta u}{f}\right), \quad \gamma = \arctan\left(\frac{m \cdot \Delta v}{f}\right) \tag{4.2}$$

Using this method, all pixels neighboring the center pixel are utilized to control the angular dependency of the light field.

5 Experiments

To ensure the procedures described above work as expected, we tested them on a light field display constructed from a smartphone featuring a 4K (3840 x 2160 pixels) screen and an array of 100 x 100 micro-lenses with a diameter of 645 µm and a focal length of 3mm each. This corresponds to a total range of possible emission angles of ±6° and an angular resolution of 0.6°.

Figure 5.1 was taken with only the determined center pixels active while the camera remained in the same position from which the

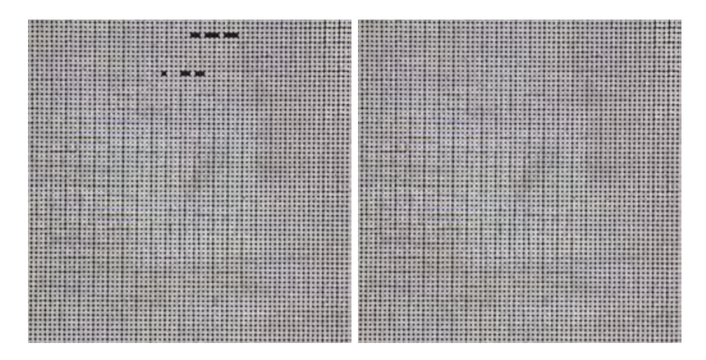


Figure 5.1: Activating only the center pixels determined by the calibration routine before (left) and after (right) error correction.

calibration was conducted. Bright micro-lenses correspond to correctly identified center pixels. After error correction, we are able to correctly address 100% of the center pixels.

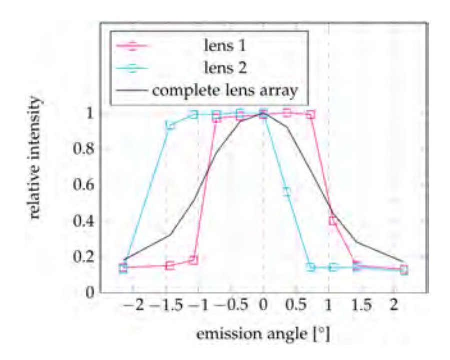


Figure 5.2: With the appropriate rendering (top), different colors are emitted into three distinct directions, each 3° apart. The images taken from the corresponding viewpoints (bottom).

Figure 5.2 shows the angular dependency of the light field display when only the center pixel is activated. The colored graphs depict two extreme cases of single lenses deviating from the targeted emission angle of 0°. They are probably caused by the integer pixel approximation which deviated most from the optimum when the

optical axis of the lenses point at pixel boundaries. The black graph represents the overall angular emission distribution, which is indeed centered at 0°.

The emission angle of $\pm 1^\circ$ at full width half maximum is caused by minor misalignment of the lens array relative to the smartphone display. Because the actual light emission takes place at an unknown distance beneath the cover glass, the correct focal distance of 3mm is difficult to adjust. For this reason and mostly for the fact that the light is not emitted from a point but a spatially extended area, i. e. the actual pixel, perfectly collimated light beams are impossible to achieve.

Figure 5.3 demonstrates that the light field display's angular selectivity can perfectly be utilized for visual inspection. We projected a spatially uniform light field consisting of three collimated beams into three distinct directions, each 3° apart and encoded them with the base colors red, green and blue. As expected, the light field display appears red, green or blue, depending on the viewing angle. Hence, it can be used to create lighting setups comparable to those shown in Figure 2.1.

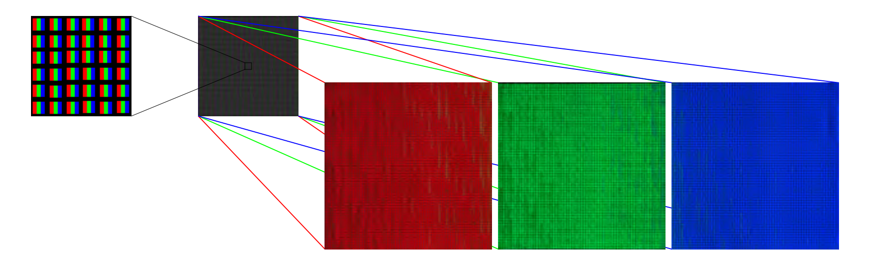


Figure 5.3: With the appropriate rendering (top), different colors are emitted into three distinct directions, each 3° apart. The images taken from the corresponding viewpoints (bottom).

The number of individual addressable angles corresponds to the number of monitor pixels behind each micro-lens. Our prototype is able to emit light into 19 x 19 = 361 different directions. Therefore, much more detailed light fields can be generated. In order to demonstrate the possibilities when controlling both spatial and angular dimensions of the light field, we simulated different perspectives of a

magic cube, cf Figure 5.4. In a second step, we rendered this synthetical light field data into the 2D-representation which is displayed on the smartphone display. The lens array then converts this into a 4D light field, which we captured with a camera from nine different viewpoints. As can be seen, the emitted light field closely resembles the original light field data.

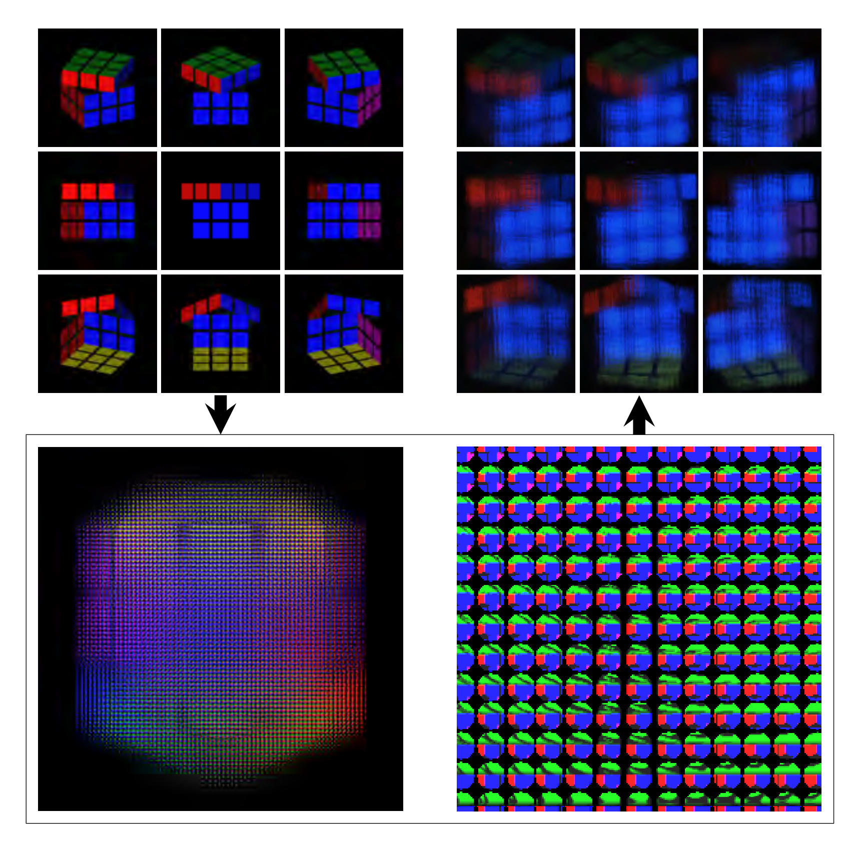


Figure 5.4: Light field rendering and imaging. Synthetic light field data of a magic cube (top left) used to render a 2D-representation optimized for the light field display used (bottom left, detailed bottom right) and the resulting light field emitted from the light field display as seen from different viewing angles (top right). The emitted images closely match the original light field data.

6 Conclusion

Because of the high number of different perspectives used in Figure 5.4, for an observer the magic cube appears as a real 3D-object floating in space. Accordingly, we are able to simulate any light source within the physical constraints of the light field display such as the spectral emission range and the pixel resolution of the monitor.

The latter ultimately limits the spatial and angular resolution of the light field display. Generally, angular dense light fields combined with a broad emission angle and a reasonable spatial resolution of the lens array can only be realized with monitors whose pixels are packed extremely dense. Currently, the limits are 807PPI at 4K resolution (3840 × 2160 pixels) [8] for smartphone displays, which are rather small, and 280PPI at 8K resolution (7680 x 4320 pixels) [9] for larger screens up to 32”.

The next step after generating light fields from synthetic data is to use recorded light fields. They can be used to realize an inverse light field illumination for visual inspection. The challenge is not only to precisely capture the light field caused by an object, but also to recode its four-dimensional, spatial-angular data to meet the physical properties of the device emitting the inverse light field.

References

1. Stemmer Imaging AG, “The Imaging & Vision Handbook,” www.stemmer-imaging.com, 2020. [Online]. Available: https://www.stemmer-imaging.com/de-de/handbuch-der-bildverarbeitung/

2. Fraunhofer IOSB. (2020, Jun.) Purity-Qualitätsprüfung transparenter Objekte. https://www.iosb.fraunhofer.de/servlet/is/5208/. [Online]. Available: https://www.iosb.fraunhofer.de/servlet/is/5208/

3. J. Beyerer, *Machine Vision : Automated Visual Inspection : Theory, Practice and Applications*, F. Puente León and C. Frese, Eds. Berlin: Springer, [2016]. [Online]. Available: http://d-nb.info/1071728458/04;http://deposit.d-nb.de/cgi-bin/dokserv?id=5282917&prov=M&dok_var=1&dok_ext=htm;http://swbplus.bsz-bw.de/bsz456265708cov.htm

4. R. Gruna, “Beleuchtungsverfahren zur problemspezifischen Bildgewinnung für die automatische Sichtprüfung,” Ph.D. dissertation, Karlsruher Institut für Technologie (KIT), 2013.
5. A. Schöch, P. Perez, and S. Linz-Dittrich, “Automated classification of imperfections and dust un small optical elements,” in *Forum Bildverarbeitung 2018. Hrsg.: T. Längle, P. L. Fernando, M. Heizmann*. KIT Scientific Publishing, Karlsruhe, 2018, pp. 161–172.
6. R. Matsubara, Z. Y. Alpaslan, and H. S. El-Ghoroury, “Light field display simulation for light field quality assessment,” in *Stereoscopic Displays and Applications XXVI*, N. S. Holliman, A. J. Woods, G. E. Favalora, and T. Kawai, Eds., vol. 9391, International Society for Optics and Photonics. SPIE, 2015, pp. 116 – 130. [Online]. Available: https://doi.org/10.1117/12.2083438
7. J. Meyer, “Light Field Methods for the Visual Inspection of Transparent Objects,” Ph.D. dissertation, Karlsruher Institut für Technologie (KIT), 2018.
8. Sony Europe B.V., “Sony Xperia XZ Premium - Technische Daten,” www.sony.de, 2020. [Online]. Available: https://www.dell.com/de-de/work/shop/accessories/apd/210-amfd#techspecs_section
9. Dell Technologies Inc., “Dell UltraSharp 32 PremierColor UltraHD 8K,” www.dell.com, 2020. [Online]. Available: https://www.dell.com/de-de/work/shop/accessories/apd/210-amfd#techspecs_section

Modulares Ringlicht für photometrische Analyse von Mikrostrukturoberflächen

A. Haider, L. Traxler, N. Brosch und C. Kapeller

AIT - Austrian Institute of Technology GmbH
High-Performance Vision Systems
Giefinggasse 4, 1210 Wien

Zusammenfassung Die Analyse von mikrostrukturierten Oberflächen erfordert praktische Analysetools. In diesem Beitrag stellen wir ein modulares Ringlicht vor, welches in einem Mikroskopaufbau integriert wird. Beide Geräte sind miteinander synchronisiert und werden durch einen Strobe-Controller angesteuert. Das Ringlicht besteht aus sechs individuell ansteuerbaren Lichtquellen, die eine Oberfläche aus unterschiedlichen Winkeln beleuchten. Das verbessert die Sichtbarkeit von Mikrostrukturen / Defekten und ermöglicht weitere Analysen mittels Photometrischen Stereo Algorithmen. Das modulare Design ermöglicht einfaches Tauschen von Lichtquellen und beispielsweise die Verwendung von Lichtquellen unterschiedlicher Wellenlänge. Diese Flexibilität macht das Ringlicht zu einem praktischen Analysetool für unterschiedliche Materialien. In diesem Beitrag wird die Konstruktion und das optische Design beschrieben und validiert. Zusätzlich zeigen wir Aufnahmen und photometrische Auswertungen von mikrostrukturierten Oberflächen.

Keywords Optische Inspektion, Ringlicht, mikrostrukturierte Oberflächen, Photometrisches Stereo, Mikroskopie

1 Einleitung

Dieser Beitrag widmet sich der Entwicklung eines Analysetools, einer modularen Ringlichtquelle samt Ansteuerungssoftware für ein Mikroskop, mit welchem beispielsweise Riblet Folien [1] untersucht

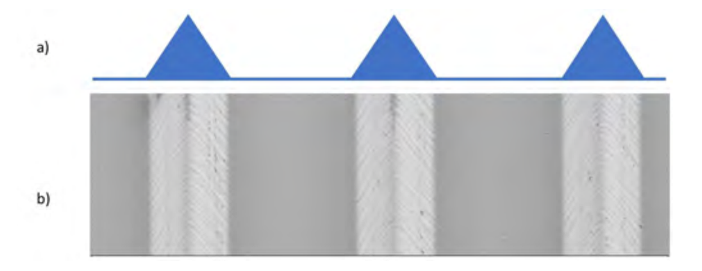


Abbildung 1.1: a) Schematische Darstellung: Riblet Profil. b) Aufnahme mit Rasterelektronenmikroskop (Aufsicht): Riblet Folie mit periodischen Strukturen in der Größenordnung von $20 - 100\ \mu m$ und Spitzen im einstelligen μm-Bereich.

werden können. Hierbei handelt es sich um Oberflächen mit periodischen Strukturen in der Größenordnung von $20 - 100\ \mu m$ (Abbildung 1.1), welche der Verringerung des Strömungswiderstands dienen. Sie werden zur Reduktion des Treibstoffverbrauchs in der Luftfahrt [2], zur effizienteren Energieerzeugung mit Windkraftanalgen [3] oder in Turbomaschinen [4] eingesetzt. Die Verringerung des Strömungswiderstands steht im direkten Zusammenhang mit der Ribletgeometrie, welche sich durch Abnutzung / Defekte verändern kann [5].

Die Qualitätssicherung erfordert ein Analysetool, bestehend aus Mikroskop und Beleuchtungseinheit, welches Strukturen im einstelligen μm-Bereich sichtbar macht, d.h., auflöst und optimal beleuchtet. Das Mikroskop [6] mit 10-facher Vergrößerung hat eine Auflösung von $0,7\ \mu m/px$ und ein Field of View von $1,624 \times 1,208\ mm^2$. Das dafür entwickelte Ringlicht besteht aus sechs fokussierten Lichtquellen und kann, im Vergleich zu Beleuchtungen mit nur einer Lichtquelle, strukturelle Defekte besser sichtbar machen [7]. Deren Reflexionseigenschaften können dazu führen, dass sie nur bei Beleuchtung aus einem bestimmten Winkel sichtbar sind (z. B. Kratzer orthogonal zur Beleuchtungsrichtung). Durch alternierende Beleuchtung aus mehreren Winkeln, können verschiedene Defekte sichtbar gemacht und komplexere Auswertungen, wie photometrische Stereo Analysen (z. B. [8]), durchgeführt werden. Unser

Analysetool, erstellt solche Aufnahmen automatisch, indem einzelne Lichtquellen mittels eines Strobe-Controllers angesteuert und mit der Mikroskop-Kamera synchronisiert werden.

Bei der Konstruktion des Ringlichts wurde auf austauschbare, modulare Lichtquellen und auf optische Standardkomponenten gesetzt. Dadurch können einzelne Lichtquellen getauscht und das Ringlicht beispielsweise mit Infrarot- oder kohärenter Laser-Lichtquellen betrieben werden. Mit einer Beleuchtungsstärke von 40 Mlx pro Lichtquelle realisiert das Ringlicht optimale Beleuchtungsbedingungen. Die hohe Beleuchtungsstärke ermöglicht kurze Belichtungs- und Strobezeiten (10 ms) und sogt daher für einen schnellen Aufnahmeprozess.

State-of-the-Art Ringlichter erfüllen die Anforderungen bezüglich Beleuchtungsintensität, Beleuchtungsrichtungen und Modularität nicht. Als Vergleich dient die Vier-Segment-Ringlichtbeleuchtung (HPR2-250SW-DV04M12-5)[1] von Computational Imaging mit einer Gesamtleistung von 46 W. Dessen Lichtquellen sind fest verbaut und beleuchten Oberflächen diffus. Das in diesem Beitrag entwickelte Ringlicht ist im Gegensatz modular konzipiert und ermöglicht im Pulsbetrieb Spitzenleistungen von 60 W pro Lichtquelle.

Abschnitt 2 beschreibt die Konstruktion, das Lichtdesign sowie die Ansteuerung des modularen Ringlichts. In Abschnitt 3 wird das entwickelte Ringlicht experimentell validiert. Abschnitt 4 zeigt Aufnahmen und Ergebnisse einer photometrischen Stereo Analyse von Riblet Oberfläche.

2 Modulares Ringlicht für Photometrische Analyse von Mikrostrukturoberflächen

Das Ringlicht beleuchtet software-gesteuert die Oberfläche aus sechs Winkeln und ergibt, gemeinsam mit einem Mikroskop [6], ein photometrisches Analysetool (Abbildung 2.1 *a*) für Mikrostrukturoberflächen. Unten beschreiben wir die modulare Konstruktion (Abschnitt 2.1), das Lichtdesign (Abschnitt 2.2) und die Ansteuerung (Abschnitt 2.3).

[1] http://www.computationalimaging.com/files/HPR2-DV04M12-5-Product-Introduction-Sheet.pdf

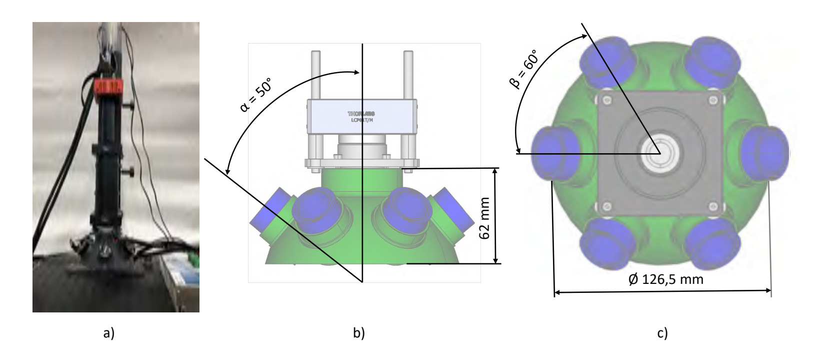


Abbildung 2.1: a) Analysetool bestehend aus Mikroskop [6] und Ringlicht. b) und c) stellen das Ringlicht, bestehend aus Dome (*grün*) und Tubes (*blau*), dar. b) Vorderansicht, Domehöhe 62 mm. c) Aufsicht, Domedurchmesser $\varnothing 126,5$ mm.

2.1 Konstruktion mit Rapid-Prototyping-Verfahren

Das Ringlicht (Abbildung 2.1) besteht aus folgenden Hauptkomponenten: (i) Schirm, (ii) sechs Tubes und (iii) sechs Lichtquellen.

Der kuppelförmige Schirm (i) ist als 3D-Druck ausgeführt (Abbildung 2.1 *b* und *c*). In dessen sechs Aussparungen wird jeweils eine Tube/Lichtquelle eingesetzt. Aufgrund der Materialwahl können die Lichtquellen mit hohen Leistungen betrieben werden, da das für den 3D-Druck verwendete Acrylnitril-Butadien-Styrol-Copolymer (ABS) im Bereich von 40°C bis 75°C formstabil und temperaturbeständig [9] ist.

Die Tubes (ii) sind ebenfalls als 3D-Druck ausgeführt. Sie werden in die Aussparungen im Schirm eingesetzt und beinhalten selbst jeweils eine Lichtquelle. Da die Tubes samt Lichtquellen einfach und schnell ausgetauscht werden können, sorgen diese für ein modulares Design.

Die Lichtquellen (iii) befinden sich in den Tubes. Sie strahlen mit einem Steigungswinkel von 40° auf die Oberfläche (Abbildung 2.1 *b*) und sind in 60° zueinander radial um die optische Achse angeordnet (Abbildung 2.1 *c*). Durch diese Beleuchtungswinkel werden die Mikrostrukturen mit hoher Sensitivität erkannt.

2.2 Lichtdesign

Das Lichtdesign umfasst sowohl das Design der Lichtführung als auch die Auswahl passender Lichtquellen. Das vorgestellte Analysetool dient der Untersuchung von Mikrostrukturen und erfordert damit die Beleuchtung von kleinen Inspektionsflächen ($1,624 \times 1,208\ mm^2$). Das Licht wird auf diese kleine Fläche fokussiert und dadurch die Beleuchtungsintensität erhört. Diffuse, im Gegensatz zur gewählten fokussierten, Beleuchtung, würde das Licht an die Umgebung abgeben und damit Mikrostrukturen nicht ausreichend beleuchten.

Die Fokussierung erfolgt durch die Kollimator-Optik, bestehend aus zwei gegengleich zueinander angeordneten asphärischen Linsen. Die erste Linse kollimiert und die zweite Linse bündelt das Licht homogen auf die Inspektionsfläche. Das Lichtdesign und die Auswahl der Linsen erfolgten mittels Optikdesign-Software[2]. Abbildung 2.2 zeigt das Ergebnis des optischen Lichtdesigns mit eingezeichnetem Strahlengang und definierten Brennweiten (16 mm bzw. 50 mm) der Linsen unter Einhaltung des Abstrahlwinkels der Lichtquellen. Als Lichtquellen dienen Leuchtdioden[3] mit einem Lichtstrom von 545 lm bei einem Betriebsstrom von $1,8\ A$, einer Leistung von 6 W, einem Abstrahlwinkel von 150° und einer Farbtemperatur von 5000 K. Da die Beleuchtungsrichtungen in einem Zeitmultiplex-Verfahren sequentiell aufgenommen werden, kann der Lichtstrom der Leuchtdioden durch Pulsen noch weiter gesteigert werden. Die Lichtquellen können bei kurzen Pulsen im ms-Bereich problemlos mit dem Zehnfachen des angegebenen Stroms betrieben werden ohne die Leuchtdioden zu überhitzen. Bei einem Strom von 18 A und einer Leistung von ca. 60 W lassen sich Lichtströme von bis zu 1911 lm beobachten.

Bei der Wahl der Leuchtdioden wurde auf eine kleine Etendue mit hohem Lichtstrom geachtet, damit die Lichtstrahlen möglichst vollständig von der, darauf angepassten, Kollimator-Optik aufgefangen werden. Das gewährleistet eine maximal mögliche Lichtausbeute.

[2] Zemax, http://zemax.com

[3] Osram Power LED: OSLON LED typ. 5000K CSSRM2.EM-MFN3-XX33-K2L1-700-R18

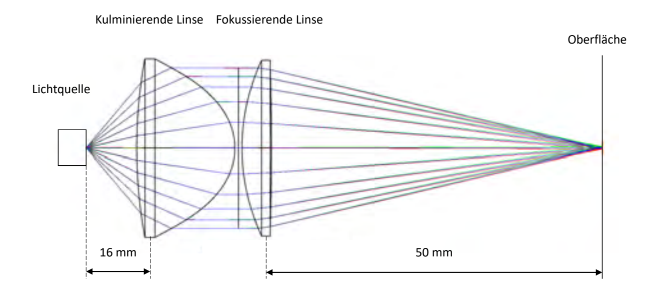


Abbildung 2.2: Entstehender Strahlengang der Kollimator-Optik durch die Verwendung von zwei gegengleich angeordneten asphärischen Linsen.

2.3 Ansteuerung

Die Ansteuerung wird mittels einer graphischen Benutzeroberfläche und einem Strobe-Controller realisiert. Die Lichtquellen sind mit dem Strobe-Controller verbunden, der diese über definierte digitale Pulse einschaltet. Ein weiterer Puls am Triggereingang der Kamera sorgt für das Starten der Belichtungszeit und das Erstellen einer Aufnahme. Die Lichtquellen werden vom Strobe-Controller über eine Konstantstromquelle versorgt. Diese verhindert Beschädigung der Leuchtdioden bei längerem Betrieb (temperaturabhängige erhöhte Stromaufnahme).

3 Experimentelle Designvalidierung

In diesem Abschnitt werden die Konstruktionsparameter des Ringlichts validiert. Die vertikalen Beleuchtungswinkel (i) wurden experimentell mit einem Light-Dome ermittelt. Durch die Messung der Helligkeit gegenüber der Stromstärke (ii) wurde die Erhöhung des emittierenden Lichts mit zunehmender Stromstärke getestet.

Der optimale Beleuchtungswinkel (i) wurde mit einem photometrischen Light-Dome identifiziert. Im Light-Dome sind 32 Lichtquellen auf drei unterschiedlichen Ebenen angeordnet. Dadurch wird erreicht, dass die Lichtquellen mit 65°, 52° und 36° auf die Oberfläche strahlen (horizontal gemessen). Aus den theoretischen Überlegungen

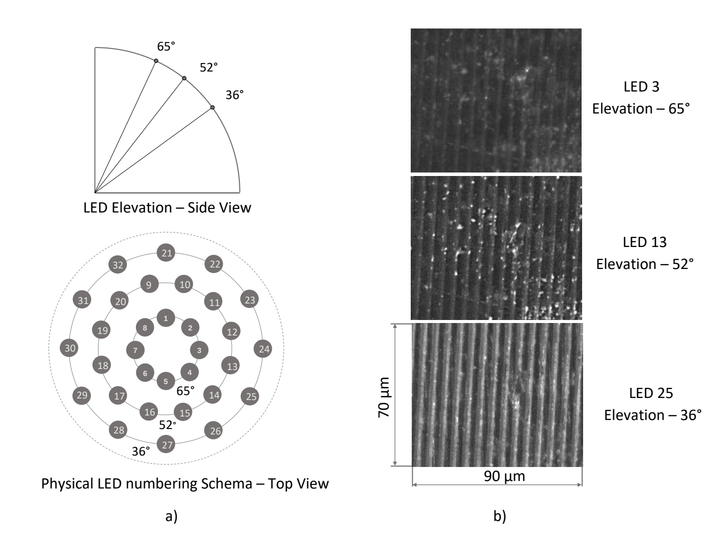


Abbildung 3.1: a) Lichtwinkel des Light-Domes: Oben rechts Beleuchtungswinkel von 65°, mittig mit 52° und unten mit 36°. b) Ausschnitte der Aufnahmen mit Beleuchtung aus 65°, 52° und 36°. Je kleiner der Beleuchtungswinkel in diesem Light-Dome, umso höher ist die Erkennbarkeit der Ribletstrukturen.

geht hervor, dass der optimale Beleuchtungswinkel im Bereich von 40° liegt, da die dreieckigen Ribletstrukturen eine Steigung von 40° aufweisen. Das zusätzlich durchgeführte Light-Dome-Experiment bestätigt die theoretischen Überlegungen: Die Struktur ist in Aufnahmen mit einer Beleuchtung aus einem Winkel von 36° am sichtbarsten (Abbildung 3.1 b, LED Elevation - 36°). Aus diesem Grund wurde der Beleuchtungswinkel des entwickelten Ringlichts mit 40° konstruktiv festgelegt.

Die Messung der relativen Helligkeit gegenüber der Stromstärke (ii) testet die Änderung des abstrahlenden Lichts mit zunehmender Stromstärke. Die Auswertung der mittleren Helligkeit erfolgt am Sensor des Mikroskopaufbaus. Abbildung 3.2 zeigt, dass die relative Helligkeit beim Betriebsstrom von $1,8\ A$ 100% beträgt - dies entspricht dem im Datenblatt angegebenen Lichtstrom von 545 lm. Wird der Strom im Pulsbetrieb weiter erhöht, lässt sich die relative Hellig-

keit um über 250% auf 351% bei 18 A steigern. Es lässt sich festhalten, dass die Leuchtdioden im Pulsbetrieb kurzfristig mit einem zehnfach höheren Strom betrieben werden können als im Datenblatt angegeben. Somit kann für mehr Helligkeit innerhalb kürzester Belichtungszeiten auf der Mikrostrukturoberfläche gesorgt werden.

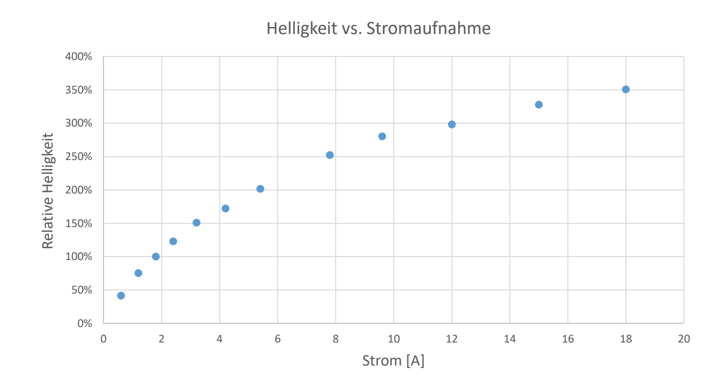


Abbildung 3.2: Darstellung der relativen Helligkeit gegenüber der Stromaufnahme der Leuchtdioden. Auf der x-Achse ist der Strom in Ampere und auf der y-Achse die relative Helligkeit in Prozent aufgetragen.

4 Ergebnisse der Oberflächenanalyse

Dieser Abschnitt zeigt Aufnahmen (Abbildung 4.1 *a*), welche mit dem vorgestellten Analysetool gemacht wurden und das Ergebnis der photometrischen Stereo Auswertung (Abbildung 4.1 *b*, Abbildung 4.2 *a* und *b*). Das Analysetool erstellt automatisch pro Lichtrichtung eine Mikroskopaufnahme eines Objekts (z. B. Riblet Folie). Nach der Kalibrierung der Lichtquellen (Lage und Intensität), kann mittels photometrischer Stereo Analyse [8], auf die Oberflächenorientierung geschlossen werden. In den berechneten Oberflächennormalen können strukturelle Defekte der Oberfläche erkannt werden. Abbildung 4.1 *b* zeigt das Ergebnis einer photometrischen Auswertung (Oberflächennormalen), berechnet aus den unterschiedlich beleuchteten Aufnahmen einer Riblet Folie, die in Ab-

bildung 4.1 *a* gezeigt werden. In diesem Beispiel sind Defekte deutlich als Unregelmäßigkeiten in der horizontalen Linienstruktur erkennbar (Abbildung 4.1, gelbe Markierungen). Obwohl die Beleuchtungswinkel des Analysetools für die Strukturen der Riblet Folien optimiert wurden, kann dieses auch zur (photometrischen) Untersuchung beliebiger mikrostrukturierter Oberflächen herangezogen werden. Abbildung 4.2 zeigt photometrische Auswertungen des Intaglio-Drucks eines 20-Euro-Scheins und einer 5-Cent-Münze. Das einfache Tauschen der Lichtquellen (z. B. Infrarot) ermöglicht außerdem die Analyse von Proben mit unterschiedlichen Materialeigenschaften.

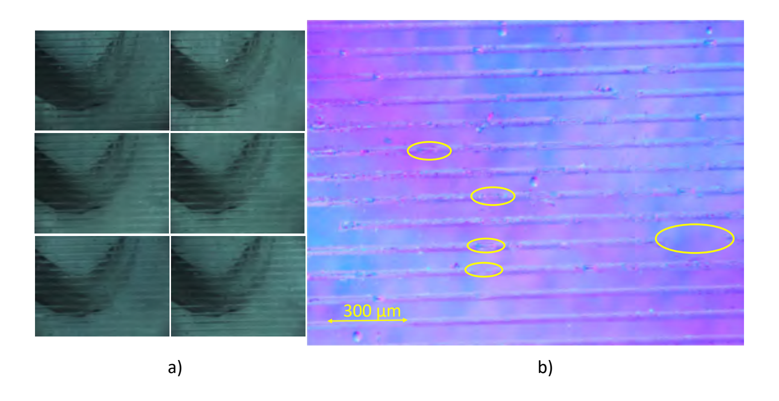


Abbildung 4.1: a) Erstellte Aufnahmen der Oberfläche und b) Darstellung der abstrahlenden Oberflächennormalen der photometrischen Stereo Analyse einer beschädigten Riblet-Oberfläche. Die Defekte sind als Abschliffe der Struktur erkennbar, einige Defekte sind gelb markiert.

5 Zusammenfassung

In diesem Beitrag wurde ein Ringlicht vorgestellt, welches zusammen mit einem Mikroskop ein praktisches Analysetool bildet. Das Lichtdesign wurde mit einer Optikdesign-Software erstellt und in zwei Experimenten validiert. Das Ringlicht zeichnet sich durch eine hohe Beleuchtungsstärke (40 Mlx), kurze Belichtungs- bzw. Strobe-

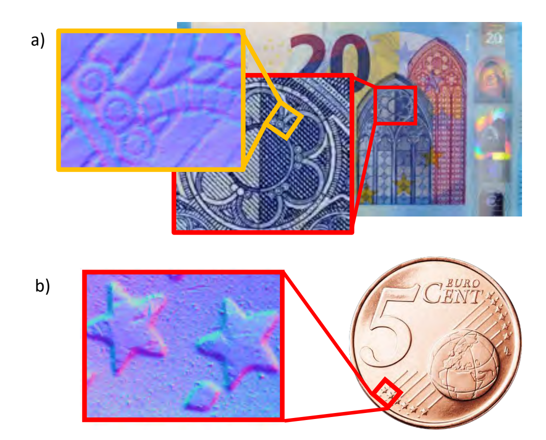


Abbildung 4.2: a) Darstellung der abstrahlenden Oberflächennormalen der photometrischen Stereo Analyse eines Intaglio-Drucks eines 20-Euro-Scheins und b) Darstellung der abstrahlenden Oberflächennormalen der photometrische Stereo Analyse einer 5-Cent-Münze.

zeiten (10 ms) und Modularität aus. Die von uns entwickelte Beleuchtung wurde mittels Rapid-Prototyping-Verfahren konstruiert und ist aufgrund von einfach austauschbaren Standardkomponenten vielseitig anwendbar. Das Ringlicht eignet sich als photometrische Beleuchtung, da mehr als drei unabhängig voneinander ansteuerbare Lichtquellen mit bekannter Lichtintensität und Lage zur Verfügung stehen. An den photometrischen Stereo Auswertungen von Riblet Folien zeigen wir, dass sich das Ringlicht für Analysen im einstelligen μm-Bereich eignet.

6 Dankaussagung

Dieser Beitrag wurde vom Resreach und Invoationsprogramm „Beyond Europe“ unter Projekt RiSPECT (Projekt Nummer 874163) gefördert.

Literatur

1. V. Stenzel, Y. Wilke, and W. Hage, “Drag-reducing paints for the reduction of fuel consumption in aviation and shipping,” *Progress in Organic Coatings*, vol. 70, no. 4, pp. 224–229, 2011.
2. P. Leitl, S. Kuntzagk, A. Flanschger, and K. Pfingsten, “Experimental and numerical investigation of the reduction in skin friction due to riblets applied on the surface of a taylor-couette cell,” in *AIAA Scitech 2019 Forum*, 2019.
3. P. Leitl, S. Schreck, C. Feichtinger, A. Flanschger, V. Stenzl, H. Kordy, Y. Kowalik, and D. Stuebing, “Riblet-surfaces for improvement of efficiency of wind turbines,” in *AIAA Scitech 2020 Forum*, 2020.
4. E. Costa, M. G. de Albeniz, S. Barberis, and P. Leitl, “Increase of compressor performance through the use of microstructures,” in *Conference on Sustainable Mobility*. SAE International, 2019.
5. W. Hage, *Zur Widerstandsverminderung von dreidimensionalen Riblet-Strukturen und anderen Oberflächen*. Berlin: Mensch und Buch Verlag, 2004.
6. L. Traxler and S. Štolc, “3D microscopic imaging using structure-from-motion,” in *IS&T International Symposium on Electronic Imaging 2019*, no. 16. Society for Imaging Science and Technology, 2019, pp. 1–6.
7. Y.-H. Tsai, D.-M. Tsai, W.-C. Li, W.-Y. Chiu, and M.-C. Lin, “Surface defect detection of 3d objects using robot vision,” in *Industrial Robot: An International Journal*, vol. 38. Emerald Group Publishing Limited, 2011, pp. 381–398.
8. D. Antensteiner and S. Stolc, “Regularization in higher-order photometric stereo inspection for non-lambertian reflections,” in *VISIGRAPP*. SCITEPRESS, 2020, pp. 253–259.
9. H. Wittel, D. Muhs, D. Jannasch, and J. Voßiek, *Roloff/Matek Maschinenelemente: Tabellenbuch*. Wiesbaden: Springer, 2009.

Automated Quantitative Quality Assessment of Printed Microlens Arrays

Maximilian Schambach[1], Qiaoshuang Zhang[2],
Uli Lemmer[2], and Michael Heizmann[1]

[1]Karlsruhe Institute of Technology
Institute of Industrial Information Technology
Hertzstr. 16, 76187 Karlsruhe

[2]Karlsruhe Institute of Technology
Light Technology Institute
Engesserstr. 13, 76187 Karlsruhe

Abstract We propose an automated evaluation pipeline utilizing both bright field light and confocal microscope images as well as multiple quality measures to quantitatively evaluate the quality of printed microlens arrays.

Keywords Computational imaging, microlens array, inkjet printing, quality control

1 Introduction

Computational imaging, combining optical and digital signal processing to extract complex information from captured light, has gained much attention in recent years—ranging from multi-camera arrays and combined depth sensors in consumer electronics such as smart phones to coded snapshot spectral imagers [1] or light field cameras [2] explored in the scientific community. Microlens arrays (MLAs), consisting of a multitude of microscopic lenses which are regularly arranged on top of a transparent substrate, play an important role in computational imaging, most prominently in compact light field cameras in which they are placed in front of the camera's

image sensor to spatially code the incident light's angular dependence.

Conventionally, MLAs are manufactured using lithographic methods such as photoresist thermal reflow [3] and nanoimprint lithography [4]. Recently however, inkjet printing of microscopic optical components such as MLAs has become more feasible and affordable, allowing for fast prototyping and production, overall decreasing prototyping cycles when developing new computational cameras.

MLAs are printed applying the Drop-on-Demand inkjet printing method, where a specific volume of optical ink is jetted from the printer's nozzles to prior-determined spots on the substrate, forming a microlens (ML). Printing a multitude of such lenses, either using multiple nozzles and/or moving the nozzle over the substrate, an MLA is printed lens-by-lens. The geometric and optical quality of both the individual lenses as well as the overall manufactured grid depend strongly on a multitude of parameters such as the surface pretreatment of the substrate, the ink composition, the nozzle voltage applied to the piezoelectric transducer, as well as the movement speed of the nozzle and resolution of the printed pattern. Finetuning and optimizing these parameters is key when printing MLAs. However, evaluation of the printed results is usually done manually by experts which is cumbersome, time consuming, and subjective. For these reasons, an automated quantitative (and thus comparable) quality assessment of such printed MLAs is needed. This automated quantitative process allows to manufacture a multitude of MLA prototypes with systematically chosen printing parameters to optimize the overall quality of the array.

To this end, we propose an automated evaluation pipeline utilizing both bright field light and confocal microscope images of the printed MLAs as well as quality measures that can be used to assess the quality of the individual lenses and the overall MLA.

2 Automated quantitative quality assessment

There are four basic geometric quantities of the MLA that one is interested in: the ML radii and sag heights, as well as the vertical and horizontal spacing of the MLs. Furthermore, detecting defects

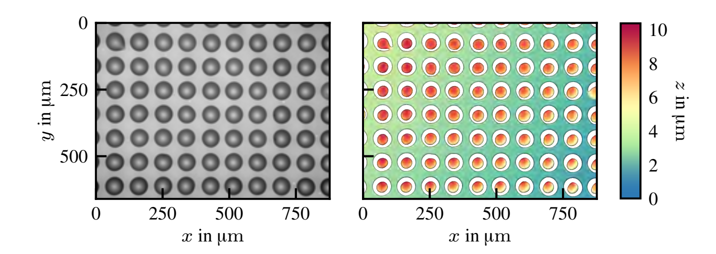


Figure 2.1: Comparison of two typical MLA measurements using a 20x lens. Left: bright field light microscope using reflected light. Right: confocal microscope (missing values are depicted in white).

in the MLA and quantifying the quality of the individual ML's shape are key to the overall assessment of the MLA quality. Finally, the back focal length of the individual MLs has to be measured.

In principle, both confocal microscopes as well as white light interferometers are well suited to measure the geometric properties of MLAs. However, both methods are incapable of providing measurements when the surface inclination is too large which is the case at the ML boundaries. Therefore, a robust measurement of the ML radii and shape is not possible with these methods. Bright field light microscope images (using reflected light) on the other hand are well suited to measure the ML shape because ML boundaries show excellent contrast precisely because of these large surface angles. A comparison of a common MLA light and a confocal measurement is shown in Figure 2.1. To measure the back focal length of the MLs, a transmitted light microscope, using a collimated light source, is well suited.

For this reason, we propose to use both bright field and confocal measurements to measure the MLA properties. Commonly, confocal microscopes offer both bright light and confocal measurements using the same optical path which makes post-capture alignment of the two measurements unnecessary (this is usually not the case for white light interferometers). In our experiments, we use a Leica DCM8 microscope with both a 20x and 50x lens. Using the 20x lens, the microscope has a lateral resolution of 0.645 µm and a vertical resolution of 1 µm whereas with the 50x lens it has a lateral resolution of

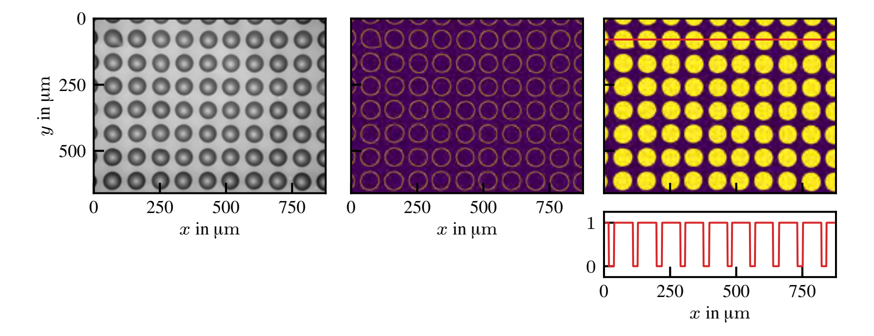


Figure 2.2: Obtaining a rough estimate of the ML radius and spacing using 1D sections of the binary image. Left: original bright field microscope image. Middle: edges detected using the Canny algorithm. Right: filled binary image and 1D section with maximum radius estimate.

0.258 µm and a vertical resolution of 0.1 µm. The MLAs are printed using a PiXDRO LP50 inkjet printer and a 10 pL cartridge. The individual MLs have a diameter of about 50–60 µm with a sag height of approximately 5 µm. Therefore, we use the 20x lens to measure all properties of the MLA except the sag height, for which a higher vertical resolution, using the 50x objective, is needed. Of course, depending on the MLAs under consideration, the chosen lenses may deviate from ours.

For each MLA a multitude of measurements is collected to increase the statistical significance of the evaluation: for the 20x magnification, we use nine confocal and corresponding bright field light microscope measurements, and three confocal and corresponding bright field light microscope measurements using the 50x lens.

2.1 Rough radius and spacing estimate

First, to bootstrap the subsequent property estimates, a rough estimate of the ML radius $\hat{r}_\mathrm{r}$ and spacing $\hat{s}_\mathrm{r}$ is obtained using a single bright field light microscope image. If multiple measurements have been collected for a single MLA, a random one is chosen. Image edges are detected using the Canny algorithm [5] followed by a filling of closed regions in the edge image. Using several hundred con-

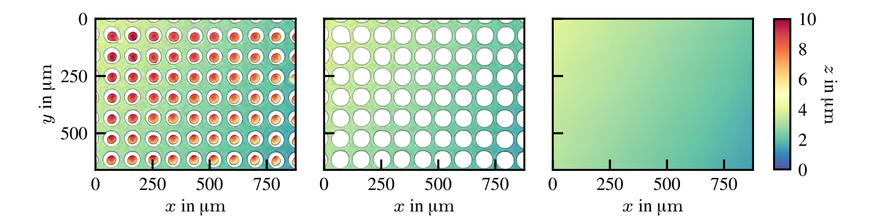


Figure 2.3: Estimating the tilt of the MLA w.r.t. the optical axis. Left: original confocal measurement. Middle: Masked confocal measurement. Right: estimated ideal MLA background plane with estimated tilt $\hat{\vartheta} = 0.09°$.

secutive horizontal 1D sections of the binary image, the radius and spacing are calculated as the maximum median number of succeeding ones (respectively zeros) of each section (compare Figure 2.2). In the case that multiple grid spacings are expected, e. g. for non-square or hexagonal grid layouts, the procedure has to be performed also for the vertical axis.

2.2 Tilt estimate

For measurements using the light microscope, the normal of the MLA and the optical axis of the microscope have to be well aligned. Misalignment leads to perspective distortions of the ideally regular grid. Hence, it will lead to systematic measurement inaccuracies when estimating the ML radius and grid spacing. To validate the MLA alignment, the flat substrate surface (on which the MLs are printed) can be used. The binary image extracted from the light microscope measurement (as described in Section 2.1) is used as a binary mask to mask out the individual MLs in the corresponding confocal measurement. To this end, a threefold binary dilation (using a fully connected 3×3 structuring element) is applied to the binary image to increase the size of the individual ML's mask. The binary mask is then applied to the confocal measurement to extract the substrate surface. Using the extracted surface, the ideal surface plane is estimated via a least-squares approximation of the measurements via the plane equation

$$z = ax + by + c \,, \tag{2.1}$$

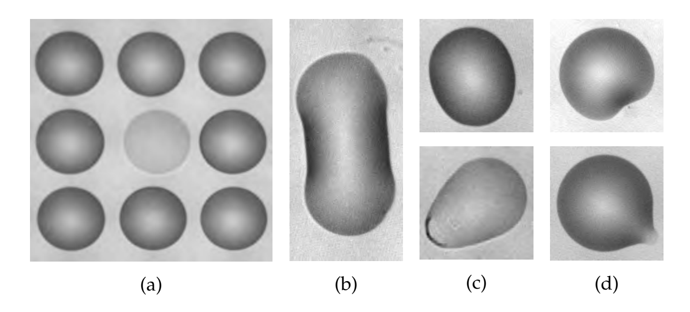


Figure 2.4: Typical shape deviations and defects in printed MLAs. (a) Missing ML. (b) Joint MLs. (c) Global shape deviation. (d) Local shape deviation.

where $\alpha = \arctan a$ and $\beta = \arctan b$ are the plane's intersection angles with the x- and y-axis, respectively. Using the estimated plane's normal vector $\hat{\mathbf{n}} = (\hat{a}, \hat{b}, 1)^{\mathrm{T}}$ and the optical axis $\mathbf{n}_0 = (0, 0, 1)^{\mathrm{T}}$, the MLA tilt angle ϑ is determined as

$$\vartheta = \arccos \langle \hat{\mathbf{n}}, \mathbf{n}_0 \rangle . \tag{2.2}$$

An overview of the procedure is shown in Figure 2.3.

When the estimated tilt is too large, the microscope tilt has to be calibrated using a tilt stage. In our experiment, we use the MLA substrate surface as a reference surface to calibrate the tilt of the MLA to be below 0.1°, however using a reference calibration mirror is also possible. In principle, when the projection matrix of the microscope is known, the estimated tilt can be used to either de-tilt the light microscope measurements or estimate an upper bound of the further MLA property estimates. However, due to the extremely narrow depth of field, a geometric calibration of the light microscope is extremely challenging.

2.3 Geometric properties estimate

Estimating the geometric properties of the MLA, one faces several challenges: First, defects in the printed MLA have to be robustly detected and taken into account when estimating the underlying regular grid's parameters. Second, the individual MLs may be deformed

and thus not perfectly circular, making the circle detection and radius estimation more difficult. Lastly, the used algorithm should not be too complex to be able to evaluate a multitude of measurements in a reasonable time.
Defects and shape deviations are common in MLA printing, in particular when the printing parameters are non-optimal and/or when the substrate surface is contaminated with dust or other particles. Common defects are missing as well as joint MLs, for which we will propose methods for detection. For the shape deviations, we roughly divide them into two classes: global and local shape deviations. Global shape deviations refer to ML shapes that are overall deviating from a perfect circular shape, for example elliptical MLs, whereas local shape deviations correspond to MLs that are overall circular with localized defects. We will introduce quality measures to quantify both types of shape deviations. For an overview of typical defects and shape deviations in printed MLAs, see Figure 2.4.

In a first step, again the edges are calculated from the light microscope measurement using the Canny algorithm. The detected edges are labeled into individual clusters using a standard labeling algorithm. Each cluster now represents exactly one ML or defect. For each cluster, the bounding box is calculated. If the larger side of the bounding box is larger than 110 % of the estimated rough diameter $2\,\hat{r}_\mathrm{r}$, the cluster is classified as a defect. This robustly detects joint lenses (which typically stretch over $4\,\hat{r}_\mathrm{r}$) as well as leaked MLs. If the larger side of the bounding box is smaller than 90 % of the estimated rough diameter, the cluster is classified as debris, containing all non-geometric defects such as dust, droplets, and scratches.

Second, the individual MLs and their radii are estimated. To this end, we propose a multi-scale extension to the circular Hough transform [6]. Since the deviation of the ML shape from a perfect circle can be quite severe (as shown in Figure 2.4), the Hough transform, applied to the original edge image, may not detect all lenses robustly. Furthermore, the accuracy of the estimated radii is limited to integer pixel values. To overcome these limitations, we perform the Hough transform on several scales $\mathcal{S} = \{s_1, s_2, ..., s_N\}$. To increase robustness against shape deviations, some scales are chosen to be smaller than one; to increase accuracy, the remaining scales are chosen to be larger than one. In our experiments we choose

$\mathcal{S} = \{0.33, 0.5, 1, 1.5, 2\}$. At each scale s_i, the edge image is calculated from the scaled light microscope measurement and the Hough transform is applied to the scaled edge image. To narrow the search space, the size of the accumulation matrix is reduced by limiting the radius range to $(1 \pm 0.1)s_i\hat{r}_\mathrm{r}$. The detected center coordinates and radii are collected together with their accumulation score. The number of detected MLs per scale decreases with larger scales: due to the shape deviations, a non-circular shape is robustly detected in the down-scaled image, however it may not reach a large accumulation score in higher scales. Therefore, starting with the lowest scale s_1, for every detected center $\mathbf{c}_i$ at scale i, a corresponding center $\mathbf{c}_{i+1}$ at the next higher scale is searched. To this end, the center $\mathbf{c}_i$ and radius r_i are projected into the higher scale:

$$\mathbf{c}_{i \to i+1} = \frac{s_{i+1}}{s_i}\mathbf{c}_i \,, \quad r_{i \to i+1} = \frac{s_{i+1}}{s_i} r_i \,. \tag{2.3}$$

Using a k-d tree-based nearest neighbor search within a unit ball of the projected radius around the projected center, the higher scale correspondent is determined. If a corresponding center is found at the higher scale, the estimated center and radius are used from that scale, if not, the current radius and center estimates are used. This procedure is repeated iteratively for every scale. The final detected centers and corresponding radii are then filtered: centers that are within a margin of $\hat{r}_\mathrm{r}$ of the image border, as well as centers that lie within the bounding box of a detected defect are neglected.

Third, using the detected centers and the initial rough spacing estimate, the grid spacing in x- and y direction is estimated, and missing MLs are detected. To estimate the spacing, following an approach similar to the grid estimation proposed by Dansereau et al. [7], a k-d tree of the final estimated centers is built. Starting with the center closest to the origin, the grid is traversed vertically and horizontally using the rough estimate of the grid basis vectors, $\mathbf{a} = (\hat{s}_\mathrm{r}, 0)$, $\mathbf{b} = (0, \hat{s}_\mathrm{r})$, in the case of a square grid. That is, the current center position $\mathbf{c}_\mathrm{curr}$ is updated by adding the corresponding grid basis vector,

$$\mathbf{c}_\text{up, horz} = \mathbf{c}_\mathrm{curr} + \mathbf{a} \,, \quad \mathbf{c}_\text{up, vert} = \mathbf{c}_\mathrm{curr} + \mathbf{b} \,. \tag{2.4}$$

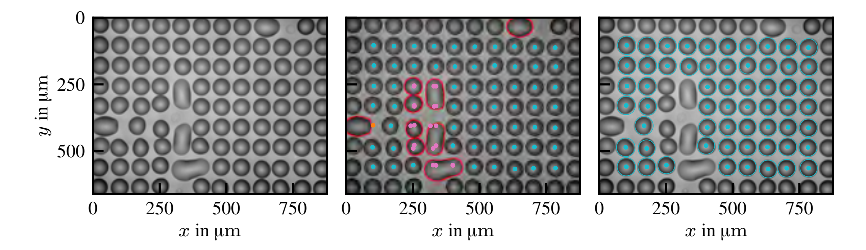


Figure 2.5: Detection results for an MLA with severe defects. Left: original bright field microscope image. Middle: detected centers (cyan), predicted centers (pink), missing MLs (orange), and detected defects (red border). Right: Detected centers and ideal circles with corresponding estimated radius.

If a detected center can be found in the neighborhood around the updated center, the found center is used as the new current center (independently for the horizontal and the vertical traverse) and the distance (vertical or horizontal) to the previous center is measured. If no center can be found, the updated center is marked as a candidate for a missing ML and used as the new current center. Having collected these distances for all MLs and multiple measurements, outliers are removed, using the median and median deviation. For example, missing MLs will lead to measured distances that are twice as large as the correct spacing and are therefore neglected. After the full grid has been traversed horizontally and vertically, the missing ML candidates are further investigated. First, since the two independent traverses may have detected the same candidates, the candidates are filtered such that there is only one unique candidate within the estimated radius. Finally, if a missing ML candidate has at least 2 detected grid neighbors and is not within the bounding box of a previously detected defect, the candidate is counted as a missing ML. An example of the detection result is given in Figure 2.5.

2.4 Microlens quality estimate

Having detected the individual MLs and estimated their radii, the geometric quality of the individual lenses is estimated. In principle, the Hough accumulation scores Q_{acc} could be used to quantify the shape quality, however theses scores are not directly comparable

Table 1: ML quality estimates for perfectly circular (top), globally deviating (middle) and locally deviating (bottom) MLs. Estimated ML centers and ideal circles with estimated radii are depicted in red. Note that the scales of the images are not identical.

ML	r/μm	$Q_{acc}/\%$	$Q_c/\%$	$Q_{cdev}/\%$	Q_{cv}
	33.4	47.02	0.84	0.97	1.15
	33.5	55.91	0.85	1.04	1.23
	32.2	31.43	5.58	6.28	1.13
	30.3	14.46	9.02	11.40	1.26
	34.5	34.87	3.56	6.54	1.84
	33.9	45.72	2.24	3.96	1.77

between different MLAs. For this reason, we propose three shape quality measures. The microlens edges have been previously labeled and clustered. For each ML, the distance d_i from every edge pixel i to the ideal ML circle (using the corresponding estimated center and radius), relative to the estimated radius, is measured. Interpreting the measured distances as realizations of a random variable $\mathtt{d}$, the following measures are defined as the sample mean, sample standard deviation, and sample coefficient of variation, respectively:

$$Q_c = \hat{\mu}_{\mathtt{d}}\,, \quad Q_{cdev} = \hat{\sigma}_{\mathtt{d}}\,, \quad Q_{cv} = \hat{\sigma}_{\mathtt{d}}/\hat{\mu}_{\mathtt{d}}\,. \tag{2.5}$$

While Q_c quantifies the overall deviation from the ideal circular shape, Q_{cdev} is well suited to measure the localization of the deviation. That is, global shape defects have a lower Q_{cdev} than local shape defects. However, larger mean deviations Q_c also in general lead to larger standard deviations Q_{cdev} which makes the values of Q_{cdev} harder to compare directly. Hence, the coefficient of variation Q_{cv} is used. A Q_{cv} close to one corresponds to circular shapes or

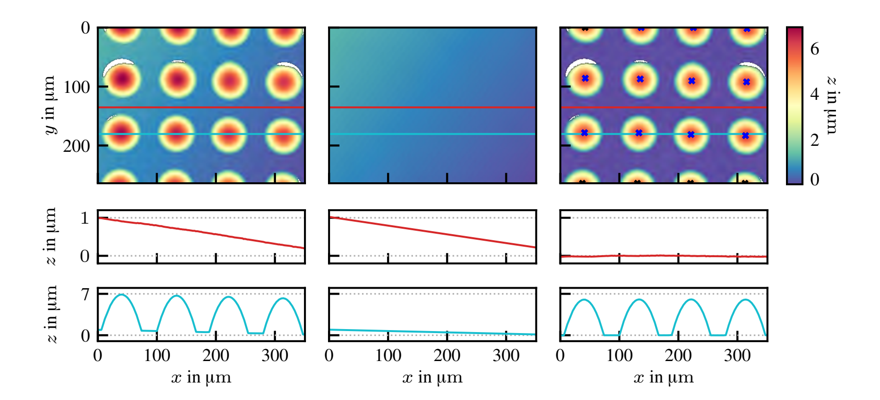


Figure 2.6: ML height estimate. Left: original confocal image. Middle: estimated ideal background plane. Right: de-tilted and zero-leveled confocal measurement with detected local maxima (blue x) and neglected outliers (black x).

shapes with a global shape defect; larger values occur when the defect is more localized. Table 1 shows some example MLs with their corresponding quality measures.

2.5 Sag height estimate

The MLA tilt and ML sag heights are estimated using the confocal and light microscope measurements at 50x magnification. In complete analogy to the procedure presented in Section 2.2 but using the 50x magnification measurements, the tilt ϑ and the offset c of the background surface are estimated. The confocal data points $\mathbf{x} = (x, y, z)$ are then zero-leveled and de-tilted,

$$\tilde{\mathbf{x}} = \mathbf{R_n}(\varphi)(\mathbf{x} - c)\,. \tag{2.6}$$

Here, the rotation matrix $\mathbf{R_n}$ is calculated from the rotation vector $\mathbf{n} = \hat{\mathbf{n}} \times \mathbf{n}_0 / \|\hat{\mathbf{n}} \times \mathbf{n}_0\|$. The individual ML sag heights can then simply be measured using the local maxima in the de-tilted and zero-leveled confocal measurement. To neglect measurements from partially imaged MLs, occuring at the image boundaries, outliers from the measured heights are removed using the median and median

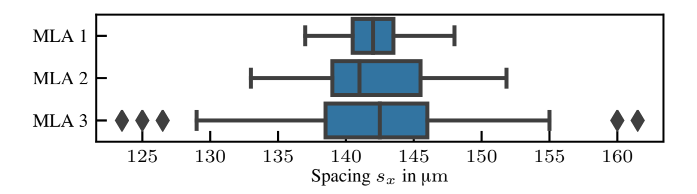


Figure 2.7: Example comparison of the measured grid spacing for three MLAs.

deviation. An example of the measurement results is depicted in Figure 2.6.

2.6 Visualization and comparison

A multitude of measurements is collected for each MLA: the radii, heights, and quality measures are measured for every individual ML, whereas the spacing is calculated pairwise. Hence, a comparison between different MLAs can be performed by either directly comparing mean and/or standard deviation values of the corresponding values or by analyzing the underlying probability distributions in more detail. For this, box or violin plots, in combination with a kernel density estimation of the data, are often used, compare Figure 2.7: while the median values of the measured grid spacings are very similar, the data of *MLA 2* and *MLA 3* are wider spread, corresponding to a less regular grid.

3 Conclusion

We have proposed and analyzed an automated evaluation pipeline utilizing both bright field light and confocal microscope images as well as multiple quality measures to automatically and quantitatively evaluate the quality of printed microlens arrays.

Acknowledgement: This work was financed by the Baden-Württemberg Stiftung gGmbH. In memory of Fernando Puente León.

References

1. G. R. Arce, D. J. Brady, L. Carin, H. Arguello, and D. S. Kittle, "Compressive coded aperture spectral imaging: An introduction," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 105–115, 2014.
2. R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report (CSTR)*, vol. 2, no. 11, pp. 1–11, 2005.
3. H. Jung and K.-H. Jeong, "Monolithic polymer microlens arrays with high numerical aperture and high packing density," *ACS Applied Materials & Interfaces*, vol. 7, no. 4, pp. 2160–2165, 2015.
4. C. Peng, X. Liang, Z. Fu, and S. Y. Chou, "High fidelity fabrication of microlens arrays by nanoimprint using conformal mold duplication and low-pressure liquid material curing," *Journal of Vacuum Science & Technology B*, vol. 25, no. 2, pp. 410–414, 2007.
5. J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.
6. R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Comm. of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.
7. D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1027–1034.

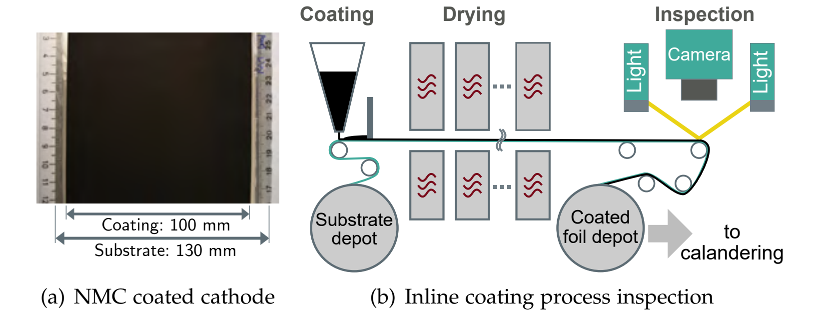
Inline battery foil inspection using strobed Photometric Stereo

Christian Kapeller, Bernhard Blaschitz, and Ernst Bodenstorfer

AIT — Austrian Institute of Technology GmbH
High-Performance Vision Systems
Giefinggasse 4, 1210 Wien

Abstract Battery technology is a key component in current electric vehicle applications and an important building block for upcoming smart grid technologies. The performance of batteries depends largely on quality control in their production process. Defects introduced in the production of electrodes can lead to degraded performance and, more importantly, to short circuits that can cause accidents. In this contribution, we propose an inspection system that can detect defects, like missing coating, agglomerates, and pinholes on coated electrodes and acquire valuable production quality control metrics, like surface roughness. By employing Photometric Stereo (PS), a shape from shading algorithm, our system sidesteps difficulties that arise while optically inspecting the black to dark gray battery coating materials. We present in detail the acquisition concept of the proposed system, and analyze its acquisition-, as well as, its surface reconstruction performance. Further, we demonstrate the acquisition results of several common defect types that arise in foil production. Our system acquires at a production speed of 500 mm/s at a resolution of 50 µm per pixel resolution.

Keywords Optical inspection, inline inspection, high-speed, electrode, photometric stereo



(a) NMC coated cathode (b) Inline coating process inspection

Figure 1.1: Illustration of a nickel manganese cobalt (NMC) cathode foil and the inline coating and inspection process.

1 Introduction

Battery technology is an important building block in the development of upcoming sustainable energy storage, energy distribution and electric mobility [1].

Electrode material is produced in the so-called coating process, in which electromechanically active material is applied onto a metal substrate foil.

A material comonly chosen for cathodes is nickel manganese cobalt (NMC) on aluminium substrate. Such cathodes exhibit a deep black texture, as shown in Fig. 6.1(a). A common choice for anodes is the dark gray colored graphite applied onto copper substrate.

For the coating process, illustrated in Fig. 6.1(b), a so-called *"slurry"*, a mixture of active material, binder material, condictive additives, and solvents, is prepared and placed into the application funnel. The slurry is applied onto the substrate with defined thickness. Following the doctor blade method [2] a blade mounted over the substrate lets slurry pass up a defined thickness. Next, the coated electrodes are dried and stored for the subsequent calandering process, where they are mechanically compressed. The goal of calandering is to improve electrode characteristics. Compression leads to more active material per volume, it homogenizes pore sizes, and reduces coating inhomogeneities. Finally, the calandered electrodes can be cut out and stacked on top of each other, interleaved with

insulating separator layers. The result can then be packaged into a final battery cell, for example in form of pouch- or prismatic cells.

An important factor influencing the electrical characteristics, and the safety of battery cells is the quality of the applied coating [3]. Ideally, the coating is finely grained and fully covers the substrate area evenly. However, especially when new kinds of coating mixes are developed, coating surfaces can deviate from this ideal conditions. A typical type of defect occurs, when the doctor blade gets clogged with agglomerates within the slurry mix, which leads to missing or unevenly applied coating behind the blade. The use of such defective electrodes in final cells degrades electrical capabilities, and, moreover, can lead to highly undesirable exothermic reactions causing harm to users [3]. Another process that can produce electrodes of suboptimal quality, is experimental battery research, when new kinds of slurry mixtures are tested. In the first case, optical quality assurance can help in ensuring that only cathodes of high quality are used for battery cells. In the second case, optical inspection can give valuable performance metrics about the quality of experimental slurry mixes.

In this work, we present an optical inspection system that can facilitate quality assurance in the coating process of anode and cathode foils that serve as building block for battery cells at production speeds of up to 500 mm/s at an optical lateral resolution of 50 μm/pix by means of photometric stereo surface reconstruction.

This paper is structured as follows. Section 2 provides an overview of existing systems for the inspection of battery foils. In Section 3 the proposed inspection system is described in detail. In Section 4 the proposed photometric stereo algorithms are explained in detail. Section 5 presents exemplary results of defects acquired with our system. Finally, Section 6 summarizes the content and provides an outlook to future work.

2 Related work

In the recent past, several optical battery foil inspection systems of various sensing modalities have been proposed. Just et al. [4] measure the infrared response of electrodes that are excited using elec-

tromagnetic radiation in order to detect applied silver particles. In contrast, our proposed system acquires material response in the visible frequency bands. Frommknecht et al. [5] combine a camera with ring-shaped illumination with a laser profilometer for defect detection. While the use of a laser profilometer allows for measuring absolute depth, its speed is limited to 500 Hz by the detectability of the laser line. Further, depth is measured only on a fraction of the foil area by the profilometer. Gruber et al. [6] employ hyper-spectral imaging and spectral ellipsometry to measure foil layer thickness while at the same time overcome specular reflections of the foil substrate. Our system, in contrast, reconstructs surfaces using a shape from shading approach.

3 Inline inspection using Photometric Stereo

In this section, we describe the proposed battery foil inspection system and its components in detail. Broadly, we can discern two subsystems, the sensor head ("Sensing & Acquisition") and the processing subsystem ("Processing & Control"), as schematically illustrated in Fig. 3.1. The sensor head is located within a coating machine and performs data acquisition and material illumination, while acquired data is processed on a PC situated in a back compartment of the machine outside of the, potentially toxic, atmosphere of the coating compartment.

3.1 Photometric Stereo acquisition and control

The sensing subsystem comprises (1) an FPGA-based controller hardware coordinating the acquisition, (2) a high-speed industrial camera, viewing at the material from top, (3) four line light sources illuminating the material from four directions, as well as, (4) a PC that coordinates image data acquisition and computes the foil surface representation.

According to Fig. 3.1, an FPGA-based controller ("Trigger Hardware"), an in-house development, ensures synchronization of material motion, control of lights, and camera acquisition. The trigger hardware translates 5 μm increments that are registered by a quadra-

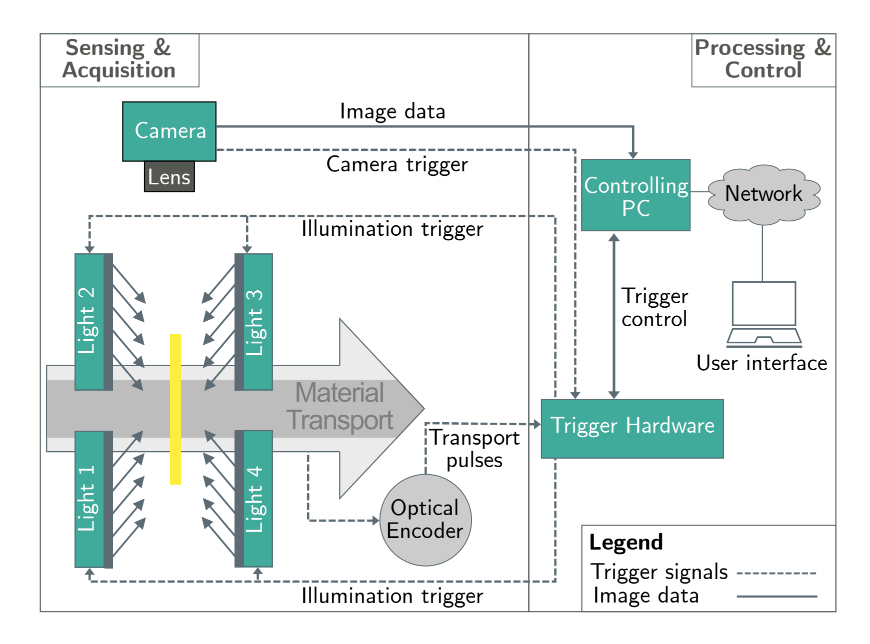


Figure 3.1: Schematic illustration of the system components. The sensing subsystem consists of a camera, viewing the electrode material from top, while it is illuminated in turn by four xposure:flash light sources for each increment measured by an optical encoder. A PC, shown in the right area, is used to setup acquisitions, and process the resulting image data, while acquisition timing is controlled by an FPGA based trigger hardware.

ture encoder to the system resolution of 50 µm increments per pixel. At each increment, the controller switches on one of the available four lights and triggers an image acquisition by the camera. This amounts to a frame trigger rate of 10 kHz at a material speed of 0.5 m/s. Fig. 6.3(a) shows the timing for switching the lights and triggering the camera. The frame period, i. e. the inverse of the frame trigger rate, corresponds to a material progress of 50 µm. Thus, as illustrated in Fig. 6.3(b), each object point (A, B, C, . . .) is acquired four times under four different illumination directions.

As illumination, four xposure:flash [7] line light sources are located in a 4-orthogonal-configuration around the camera's field of view. Each light source contains a linear array of white high-power

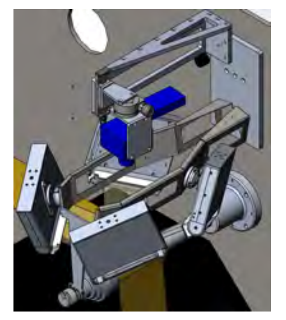
LED's that allow fast strobing. The light sources are mounted at 45° rotation with respect to the transport direction, so to deliver high quality control data for defects that often occur in transport direction. They are mounted with a polar angle of 55°, as shown in Fig. 6.2(a). This angle was experimentally determined to be optimal for this type of material. We use four light sources due to the increased surface reconstruction stability [8] compared to the three lights required for determining three dimensional surface normal vectors. In order to have enough light for a proper signal, the light sources are strobed at a frequency of 10 kHz, which is only a small fraction of their maximum strobing frequency of 600 kHz. The irradiance in the object plane, generated by a single line light, is approximately 500.000 lx.

The camera, model Mikrotron eoSens 4CXP, is configured with a 12 mm lens to exhibit a field of view (FOV) of 116 mm in horizontal, and 200 μm in vertical direction. The FOV was chosen to acquire the whole width of the material and as well as the transition from substrate to the material. At 2336 pixels sensor width this amounts to an approximate resolution of 50 μm/px. The use of a multi-line acquisition regime enables observation of the same material positions illuminated by multiple light sources. As the material is continually moving, the obtained images are shifted by 0 to 3 pixels in transport direction for lights 1 to 4, for registration (see Fig. 6.3(b)).

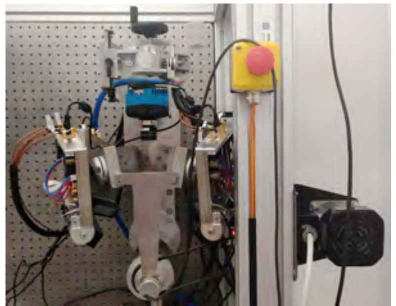
As an AIT internal project we have constructed and successfully tested a system prototype in our laboratory. Fig. 3.2 shows construction drawings, as well as an image of the prototype system including a motorized roll simulation that allows us to thoroughly test system performance under various operating speeds.

3.2 Data processing

An industrial-grade PC is used for configuration of the acquisition subsystem and processing af acquired image data. Image data is acquired via the CoaxPress interface from the camera. Photometric Stereo results are processed, and stored to harddisk for further analysis. A graphical user interface is provided in order to enable an operator to review the results in real-time. As the PC is located within the machine, its user interface is remotely accessible via Ethernet-based networking.

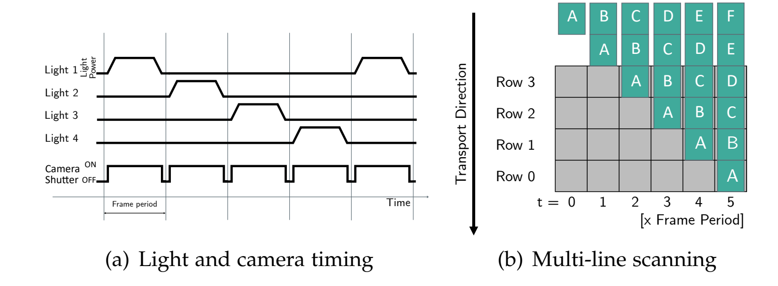


(a) Construction View



(b) Machine simulation

Figure 3.2: System illustrations of (a) construction drawings, and (b) the system prototype with attached roll simulator operating in our laboratory environment. On top, the camera can be seen, while in middle area high-speed xposure:flash light sources are visible focusing on the roll in the lower region, which is driven by a motor on the right.



(a) Light and camera timing

(b) Multi-line scanning

Figure 3.3: Illustration of the system's acquisition and light timing and System timing

4 Photometric Stereo Processing

The dark texture of the battery material, black for NMC coated cathodes to dark gray for anode foils, impedes direct optical intensity analysis. Either large amounts of light need to be used for illumination, which increases cost, or long camera exposure times need to be used, which limits the speed of the coating line. Further, 3D reconstructions can aid the quantitative assessment of electrode qual-

ity. For this reason, we analyze defects based on the reconstructed surface geometry of the material. To this end we perform surface reconstruction using Photometric Stereo [9], an shape-from shading algorithm, that is well suited for the observed diffuse material.

PS employs the Lambertian assumption of perfect diffuse material and infinitely distant, parallel light rays, and reconstructs surfaces based on observed light intensities of light reflected from surface points illuminated under several illumination angles. We compute surface normals and albedo from acquired image data that correspond to the four illumination directions using our PS algorithm [9]. In the following, we concisely summarize the method for the reader's convenience.

We determine surface normals $N_{i,j} \in \mathbb{R}^3$ and albedo $\rho_{i,j} \in \mathbb{R}$ on a discretely sampled domain of $M \times N$ pixels dimension, from n acquisitions $I_{i,j} \in \mathbb{R}^n$ illuminated by light sources of known direction $L \in \mathbb{R}^3$ relative to material surface. The matrix $M_{i,j} = \rho_{i,j} N_{i,j}$ represents surface normal vectors scaled by albedo at each location. From the known light directions $L = [X, Y, Z]$ with $X = [x_1, \ldots, x_n]$, $Y = [y_1, \ldots, y_n]$ and $Z = [z_1, \ldots, z_n]$, we construct a polynomial $P \in \mathbb{R}^{n \times 10}$ such that:

$$
\begin{aligned}
P &= [P_2, P_1, P_0], \text{with} \qquad (4.1)\\
P_2 &= [X \odot X, Y \odot Y, Z \odot Z, X \odot Y, X \odot Z, Y \odot Z],\\
P_1 &= [X, Y, Z],\\
P_0 &= [1],
\end{aligned}
$$

where $\odot$ represents the Hadamard product, P_2 denotes 2nd order basis functions, P_1 denotes surface normal vectors, and P_0 being a vector of length n modelling ambient illumination.

We determine surface normals, scaled by albedo $M_{i,j}$ using the following Tikhonov regularized model that can be solved using conjugate gradient descent.

$$
\min_{M_{i,j}} \frac{1}{2} \| P \cdot M_{i,j} - I_{i,j} \|^2 + \lambda \| \Gamma \cdot M_{i,j} \|^2 \qquad (4.2)
$$

Here $\Gamma \in \mathbb{R}^{7 \times 10}$ denotes an identity matrix for $[P_2, P_0]$. A a scalar biasing parameter λ steers the model to be explained foremostly by the coefficients in P_1 containing the surface normal components.

By solving equation 4.2 we can retrieve surface normals and albedo from $M_{i,j}$ such that:

$$\rho_{i,j} = \sqrt{M_{i,j,1}^2 + M_{i,j,2}^2 + M_{i,j,3}^2} \tag{4.3}$$

$$N_{i,j} = \frac{M_{i,j}}{\rho_{i,j}} \tag{4.4}$$

Note, that in this application, we rely on regularization to successfully solve the model which is in principle underdetermined, using only four lights. We choose the regularized model because reconstructed surface normals are less prone to large scale surface pertubations [9]. Subsequently, we generate a depth map from surface normals N by normal integration using the the method of Frankot and Chellappa [10].

5 Experimental results

In this section, we present the qualitative results obtained by acquiring data of deliberately defective anode and cathode foils provided by researchers from AIT's on–premises coating pilot line facility. The samples have been specifically selected to provide a good overview of real-world defects that can occur in foil coating, such as missing coating, coating inhomogeneities, pinholes, agglomerations, cavities and cracks.

Our dataset comprises of two black colored nickel manganese cobalt (NMC) cathodes and two graphite anodes of dark gray texture. Coating was applied with a width of 100 mm onto substrates of approximately 130 mm width. The substrate's thickness is 20 μm, whereas, the applied coating thickness ranges from 30 to 450 μm. The samples were acquired at a speed of 500 mm/s. The results were obtained using the processing pipeline described in Section 4.

The most obvious type of defect is *missing coating*, as illustrated in Fig. 5.1 (a,b). In the present samples it is caused by agglomerates clogging the doctor blade and preventing new slurry to pass the blade in those regions. Sometimes these pollutions break free after a while, as can be seen in Fig. 5.1 (a). Electrodes with missing coating are unfit for use in cells. Large scale missing coating is obviously

visible in raw image data, small blade obliterations, however, may not reach down to the substrate material.

Such defects can be called *coating inhomogeneities*. An example can be seen in Fig. 5.1 (c). Coating inhomogeneities can cause, among others, degraded cell capacity [3], and can be mitigated to some extent by subsequent calandering. Inhomogeneities are hard to detect optically, especially on the black NMC material. They are, however, visible in surface normals and the derived depth map.

Pinholes are small diameter pores, depicted in Fig. 5.1 (c), are caused by small air bubbles bursting in the drying process and can reach down to the substrate. Pinholes can be caused by inadequate slurry mix, or drying parameters [3]. They are best visible in depth maps and invisible in raw image data. Note, that the shown sample additionally exhibits small dark spots that can, but don't always coincide with pinholes. Slurry containing large air bubbles can leave *cavities* or void areas, as shown in Fig. 5.1 (d). This sample further contains *cracks* in the coating area.

While all the previously discussed defects are variations of missing active material, *agglomerates* constitute of excess material, as shown in Fig.5.1 (d). Agglomerates can be caused by an incomplete slurry mix [3] and can potentially damage the calandering roll, or if not crushed there, pierce the separator foil in final cells.

Another observable coating property is the coating's *surface roughness*. Rough coating can damage the mechanical press in the calandering process following the coating step. Examples for low roughness is shown in Fig. 5.1 (a,b), while a sample of high roughness can be seen in Fig. 5.1 (c).

6 Conclusions and future work

In this paper we have presented an inline inspection system for battery foils that can acquire 2.5D images at speed of up to 500 mm/s with a lateral resolution of 50 µm/px. We achieve this performance using tight coupling of transport movement, interleaved strobing of four line lights and image acquistion using an FPGA-based controller. By employing Photometric Stereo surface reconstruction, our system is capable of visualizing fine surface details. After briefly

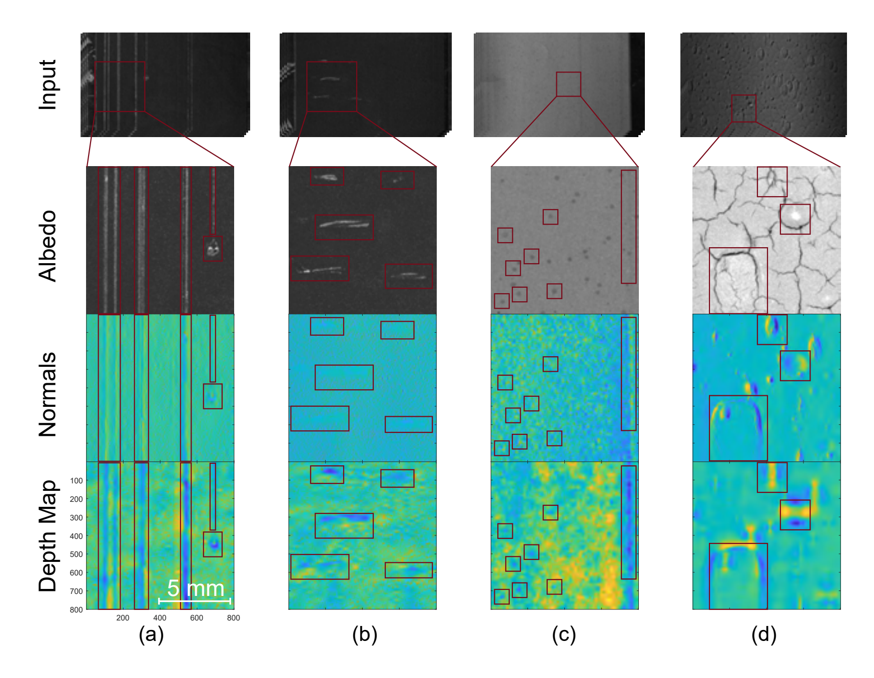


Figure 5.1: Coating defects occuring in two black NMC coated cathodes (a,b) and two dark gray graphite anodes (c,d). Defects are marked in red.

summarizing the state of the art, we have described the mechanical and optical sensing components in detail. Further, we have described our Photometric Stereo algorithm that is capable of visualizing fine surface details and defects in electrode material. Finally, we have presented qualitative results of several common foil defects in a foil data set obtained from an experimental battery production facility.

In the future, we will improve the system, so to achieve a speed of 2 m/s, while at the same time increasing the resolution to 10 µm/px, as part of the 3beliEVe project. Further, we will integrate machine-learning-based defect classification for electrodes into our system.

Acknowledgements

This work has received funding from the European Union's H2020 research and innovation program in the context of the 3beliEVe project under Grant Agreement no. 875033. Further, we would like to thank Katja Fröhlich, Lukas Neidhart and Andreas Gigl from AIT's center for Low Emission Transport for providing the electrode samples used in this work.

References

1. European Commission, "Europe on the move - Annex 2: Strategic Action Plan on Batteries," pp. 1–10, 2018.
2. H. Yang and P. Jiang, "Large-scale colloidal self-assembly by doctor blade coating," *Langmuir*, vol. 26, no. 16, pp. 13 173–13 182, 2010.
3. D. Mohanty, E. Hockaday, J. Li, D. K. Hensley, C. Daniel, and D. L. Wood, "Effect of electrode manufacturing defects on electrochemical performance of lithium-ion batteries: Cognizance of the battery failure sources," *Journal of Power Sources*, vol. 312, pp. 70–79, 2016.
4. P. Just, L. Ebert, T. Echelmeyer, and M. A. Roscher, "Infrared particle detection for battery electrode foils," *Infrared Physics and Technology*, vol. 61, pp. 254–258, 2013.
5. A. Frommknecht, M. Schmauder, L. Boonen, and C. Glanz, "Automated inline visual inspection and 3D measuring in electrode manufacturing," in *Optical Measurement Systems for Industrial Inspection XI*, no. June. SPIE, jun 2019, p. 66.
6. F. Gruber, P. Wollmann, B. Schumm, W. Grählert, and S. Kaskel, "Quality control of slot-die coated aluminum oxide layers for battery applications using hyperspectral imaging," *Journal of Imaging*, vol. 2, no. 2, pp. 1–11, 2016.
7. E. Bodenstorfer, "Ultra-schnell gepulste LED-Beleuchtung öffnet neue Dimension für optische Oberflächen-Inspektion," in *Talk: Scientific Vision Days 2018, Stuttgart; 05.11.2018 - 08.11.2018*, 2018.
8. O. Drbohlav and M. Chantler, "On optimal light configurations in photometric stereo," *IEEE International Conference on Computer Vision*, vol. II, pp. 1707–1712, 2005.

9. D. Antensteiner and S. Štolc, "Regularization in Higher-order Photometric Stereo Inspection for Non-Lambertian Reflections," in *Proc. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2020, pp. 253–259.
10. R. T. Frankot and R. Chellappa, "A Method for Enforcing Integrability in Shape from Shading Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 439–451, 1988.

Fusion of Sequential Information for Semantic Grid Map Estimation

Frank Bieder[1], Muti Ur Rehman[2], and Christoph Stiller[2]

[1] FZI Forschungszentrum Informatik, Mobile Perception Systems Department,
Haid-und-Neu-Straße 10-14, 76131 Karlsruhe
[2] Karlsruhe Institute of Technology, Measurement and Control Systems,
Engler-Bunte-Ring 21, 76131 Karlsruhe

Abstract In this work, we improve the semantic segmentation of multi-layer top-view grid maps in the context of LiDAR-based perception for autonomous vehicles. To achieve this goal, we fuse sequential information from multiple consecutive lidar measurements with respect to the driven trajectory of an autonomous vehicle. By doing so, we enrich the multi-layer grid maps which are subsequently used as the input of a neural network. Our approach can be used for LiDAR-only 360° surround view semantic scene segmentation while being suitable for real-time critical systems. We evaluate the benefit of fusing sequential information based on a dense ground truth and discuss the effect on different semantic classes.

Keywords Autonomous driving, sensor data fusion, semantic grid map estimation.

1 Introduction

Environmental perception is a crucial task for many applications in robotics and mobile systems. This is particularly true for highly dynamic environments in which human life is at stake, such as urban scenarios. In these situations, autonomous driving systems heavily rely on a robust and accurate environment interpretation and scene understanding. Semantic segmentation plays a key role in efficient,

meaningful and holistic scene representation. With the advent of deep convolutional networks the task has received a lot of attention in the last few years and has shown significant improvements. Many well-developed network architectures are tailored to the image domain due to the data shortage in other domains.

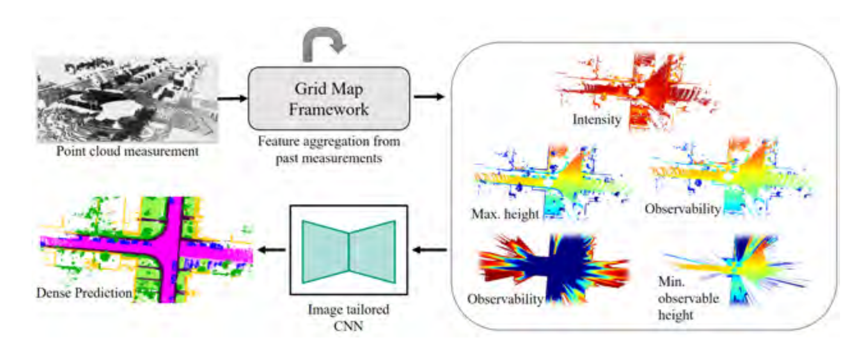


Figure 1.1: System overview including all input and output grid map types. By using our grid map framework we transform lidar measurements into a multi-layer grid map representation. The multi-layer grid maps are processed by an image-tailored CNN to predict semantic grid maps.

Recently, Behley et al. [1] published SemanticKITTI, the first large scale publicly available dataset which provides semantic segmentation for lidar measurements. The publicly available data consists of more than 23.000 single shot lidar measurements with a point-wise annotation distinguishing 28 semantic classes. By doing so the authors also provide information about moving and non-moving objects for classes like vehicle or motorcycle. In a recent work, we [2] consider the transformation of lidar point clouds into a top-view grid map representation to approach an efficient top-view segmentation of lidar measurements. The structured representation of grid maps can be utilized by applying efficient, well-developed CNN architectures from the image domain. In contrast, neural networks which operate on unstructured point clouds often lack real-time capability.

A further advantage of the grid map representation is that it is well-suited for sensor fusion applications. For instance, Nuss et al. [3] fuse radar and laser measurements to estimate the dynamic state of grid cells. Furthermore, Richter et al. [4] used grid maps as

a common fusion structure for semantic information and different range measurements. Besides the information fusion from different sensors, grid maps can also be used to fuse sequential measurement data from one sensor [5]. Another interesting work in this direction was done by Wirges et al. [6] by training a neural network to estimate dense multi-layer grid maps from single shot measurements. The paper shows that this enrichment is improving the performance of object detection algorithms.

This work investigates the fusion of sequential lidar measurements in multi-layer grid maps in the context of top-view semantic grid map segmentation.

2 Contribution

The presented work extends the basic ideas of [2] by making necessary improvements and introducing a fusion concept which replaces the single-shot approach and allows the use of sequential information. The following overview points out the main contributions of the paper:

- We extent our grid mapping framework so that it is capable of combining information from multiple point clouds into one set of grid maps. For each layer we implement a tailored fusion strategy.
- We perform semantic grid map estimation using multi-layer grid maps with accumulated features from the current and past lidar measurements.
- We report the benefit of feature accumulation in multi-layer grid maps for the task of semantic segmentation. By doing so, we evaluate the improvements on a dense semantic ground truth layer.

3 Multi-Layer Grid Maps

This section provides information about the generation and definition of our multi-layer grid maps.

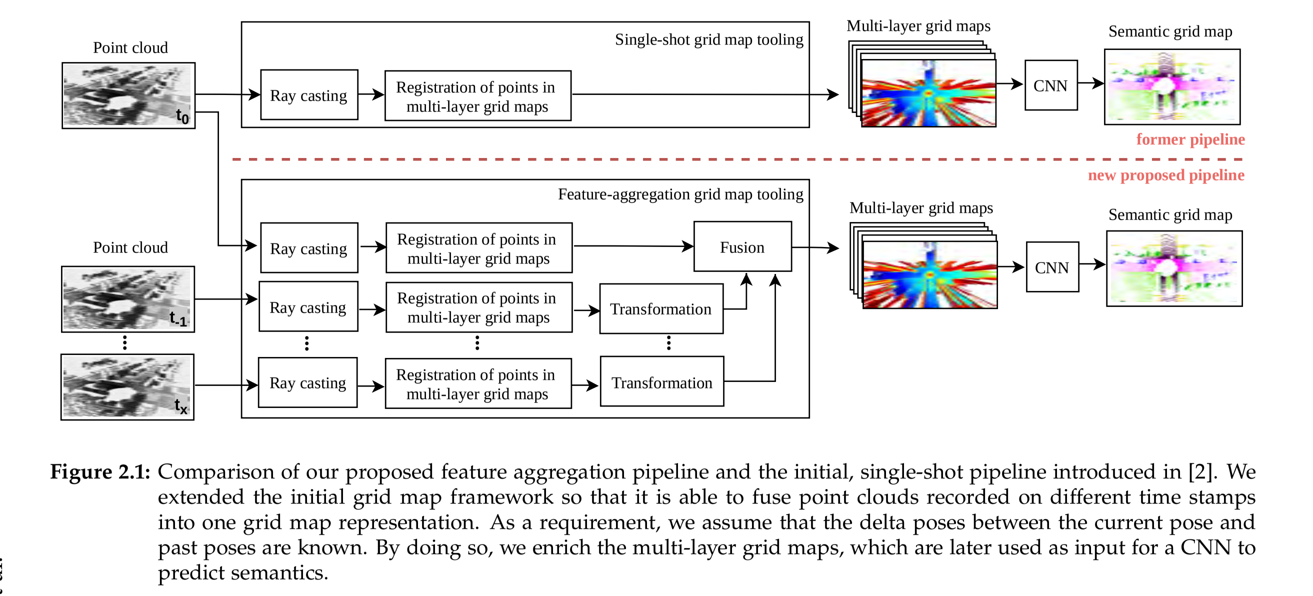


Figure 2.1: Comparison of our proposed feature aggregation pipeline and the initial, single-shot pipeline introduced in [2]. We extended the initial grid map framework so that it is able to fuse point clouds recorded on different time stamps into one grid map representation. As a requirement, we assume that the delta poses between the current pose and past poses are known. By doing so, we enrich the multi-layer grid maps, which are later used as input for a CNN to predict semantics.

Definition of Layers

Our multi-layer grid map input consists of five layers, which store the following features for each grid cell: The mean intensity, the maximum detected height, the minimum detected height,the observability representing the amount of rays through each cell and the minimum observable height with respect to all rays which crossed the cell. The first three layers only carry information in grid cells in which a lidar point is allocated. The information of the last two layers is extracted by casting rays between the sensor origin and the point detections to obtain dense layers in the observable area. In order to facilitate parallel computation and account for geometric sensor characteristics, all layers are first computed in polar coordinates and subsequently remapped into a cartesian coordinate system. An example for each layer can be found in figure 1.1.

Label Set and Data Set Split

We choose the label set and re-mapping strategy according to [2], but further combine the two classes rider and two-wheeler as they are hard to separate in the top view representation. This leads us to the following set of semantic classes: vehicle, person, two-wheel, road, side-walk, other-ground, building, pole/sign, vegetation trunk terrain. The sequences 0-7 and 9-10 of semanticKITTI are used to train the networks and the evaluation is conducted on sequence 8.

Grid Resolution and Sensing Range

The grid cell resolution is set to 10cm $\times$ 10cm. The region registered in one grid map is chosen to be 100m $\times$ 50m with the sensor located in the middle of the gird map. The grid maps are rotated such that the ego vehicles driving direction points to the right of the grid map.

Feature Aggregation

For the fusion process we collect point clouds from past time stamps, cast them to the grid map representation and transform them in the coordinate system of the current vehicle pose. We only choose past

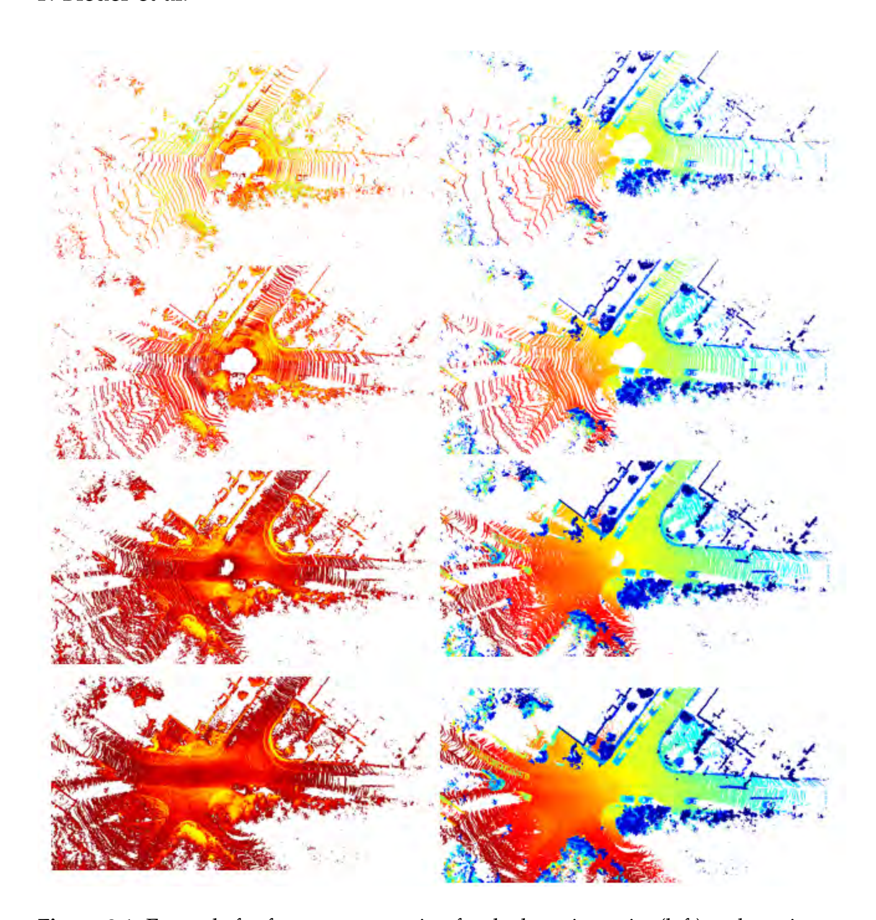


Figure 3.1: Example for feature aggregation for the layer intensity (left) and maximum detected height (right). The first row shows a single shot example, the second row 3 fused frames, the third row 10 fused frames and the last row 20 fused frames.

time stamps to have a causal system which could be applied in a similar fashion on a real-world system like an autonomous car. In order to be able to transform the past measurements into the coordinate system of the current grid maps highly accurate vehicle poses are required. We experienced that the poses of SemanticKITTI are superior of the original KITTI poses [7] and hence, use the former.

A unique fusion strategy is implemented for each layer. Regarding the intensity we calculated the average value for each grid cell considering all available measurements. In contrast we calculate the maximum value for the layer maximum detected height and the minimum value for the layers minimum detected height and minimum observable height. For the observability layer we accumulated the number of rays from each available measurement.

As the computation time for the grid mapping increases with an increasing batch size of point clouds, the number of fused measurements has to be well considered. Hence, we conduct and compare experiments with different point cloud batch sizes. An advantage of this approach is that the computational effort of the neural network does not increase by the accumulation of multiple measurements in the input grid maps.

Semantic Ground Truth

We create a dense semantic ground truth as it is described in [2]. After accumulating the semantic information of all surround poses we register the most likely pose within each grid cell. Here, we do not limit the amount of measurement but select all poses within a given radius for the fusion of semantic information.

4 Experiments

For each experiment we used all five grid map layers and optimized the network using the densely generated ground truth.

We conduct experiments comparing different state-of-the-art deep learning architectures, tailored for image processing. In this paper, all reported experiments are conducted using one architecture: the Deeplab framework with the Xception backbone [8]. We train the

networks using the full image resolution, a batch size of 2 and about 300.000 traing iterations. Besides the single shot experiments we present results for 3, 5, 10 and 20 accumulated frames.

5 Evaluation

We evaluate our experiments using the novel SemanticKITTI data set. Our models are trained to predict 11 classes which are particularly relevant for urban scene understanding. In this paper we choose a dense ground truth which also takes the network's prediction for cells without a detection into account.

Table 1: Class-wise evaluation using a dense semantic top view ground truth based on the 8 sequence of the semanticKITTI data set

frames	vehicle	two-wheel	pedestrian	road	sidewalk	parking	building	pole/sign	vegetation	trunk	terrain	**overall**
1	0.364	0.000	0.000	0.826	0.461	0.004	0.574	0.093	0.525	0.053	0.583	0.321
3	0.366	0.000	0.000	0.826	0.470	0.105	0.555	0.113	0.579	0.051	0.591	0.332
5	**0.392**	0.000	0.000	0.820	**0.487**	0.089	0.580	**0.138**	0.611	**0.064**	**0.647**	**0.348**
10	0.389	0.000	0.000	0.827	0.480	**0.128**	0.581	0.120	0.622	0.049	0.629	**0.348**
20	0.377	0.000	0.000	**0.831**	0.472	0.119	**0.583**	0.124	**0.631**	0.060	0.626	**0.348**

The quantitative evaluation is based on the *Intersection over Union* (IoU) [9]. The *mean Intersection over Union*, mIoU, is determined by

$$\text{mIoU} = \frac{1}{|K|} \sum_{k \in K} \text{IoU}_k \tag{5.1}$$

where $|K|$ is the the labelset's cardinality and the per-class IoU_k is calculated by

$$\text{IoU}_k = \frac{T_{P_k}}{T_{P_k} + F_{P_k} + F_{N_k}}, \tag{5.2}$$

with k being one of 11 classes. The quantitative results are shown in Table 1. In figure 5.1 some qualitative results are displayed.

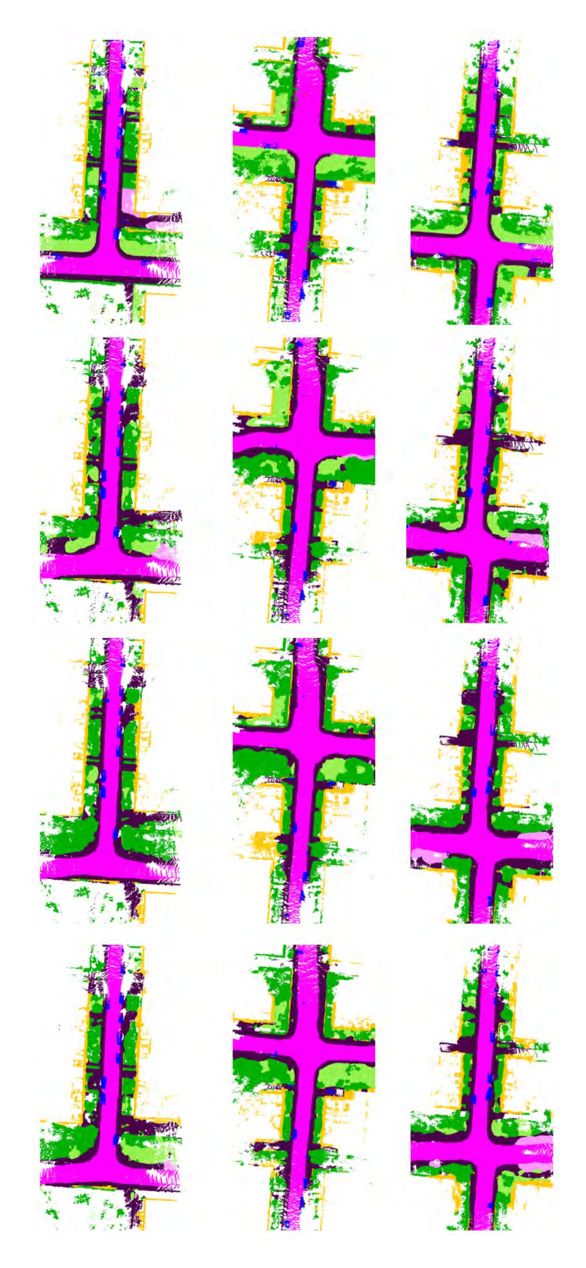


Figure 5.1: Comparison of the qualitative results of three different scenes. The first column shows the inference based on single shot grid maps, the second column with 3 frames fused, third with 10 times fused and the last column is showing the ground truth.

6 Discussion

The experiments show that improvements can be achieved by the aggregation of past measurements. The greatest benefit can be obtained for the classes terrain, trunk and vegetation, parking and for pole/sign. However the improvements of additional feature aggregation seem to stagnate if more than 5 measurements are fused. We can also review that even with the feature aggregation the classes pedestrians and two-wheel can not be semantic segmented using the multi-layer grid maps. Here we have no improvement compared to the original paper.

7 Conclusion

We propose a framework to fuse information from sequential lidar measurements in a multi-layer grid map representation. Our experimental evaluations show the benefit of our approach in comparison to a formerly introduced single-shot method. While we review that an aggregation of past measurements brings a benefit, we also show that adding more past measurements only improves the performance to a certain extent.

References

1. J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.
2. F. Bieder, S. Wirges, J. Janosovits, S. Richter, Z. Wang, and C. Stiller, "Exploiting Multi-Layer Grid Maps for Surround-View Semantic Segmentation of Sparse LiDAR Data," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2020.
3. D. Nuss, T. Yuan, G. Krehl, M. Stuebler, S. Reuter, and K. Dietmayer, "Fusion of laser and radar sensor data with a sequential Monte Carlo Bayesian occupancy filter," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2015.

4. S. Richter, S. Wirges, H. Königshof, and C. Stiller, "Fusion of range measurements and semantic estimates in an evidential framework," *tm - Technisches Messen*, 2019.

5. S. Wirges, C. Stiller, and F. Hartenbach, "Evidential Occupancy Grid Map Augmentation using Deep Learning," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018.

6. S. Wirges, Y. Yang, S. Richter, H. Hu, and C. Stiller, "Learned enrichment of top-view grid maps improves object detection," in *IEEE Conference on Intelligent Transportation Systems (ITSC), Proceedings*, 2020.

7. A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

8. L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

9. M. Everingham, S. M. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, 2014.

Light Field Reconstruction using a Generic Imaging Model

David Uhlig and Michael Heizmann

Karlsruhe Institute of Technology,
Institute of Industrial Information Technology,
Hertzstraße 16, 76187 Karlsruhe, Germany

Abstract Light field cameras play an increasingly important role in computer vision and optical metrology. However, due to their complex design, their calibration is very difficult and usually precisely adapted to the respective light field camera type. We present a method that extracts a light field from an arbitrary light field imaging system without knowing and without modelling the internal optical elements. We calibrate the camera using a generic calibration procedure, transform the obtained set of rays into an equivalent light field representation and finally, reconstruct a rectified light field from the irregularly sampled data. Experimental results validate the method and demonstrate that the geometrical structure of the light field is preserved by an adequate rectification.

Keywords Light field, decoding, rectification, generic camera

1 Introduction

The light propagating in space contains a variety of different information. However, when an image is taken with a classic camera, a large proportion of the information contained in the light is lost due to the projection. Computational cameras can encode information that is not available using conventional cameras. The additional modification of the camera can be used to extract useful information from the raw data apart from only the intensity-based colored image of the scene. In recent years, research on light field cameras (plenoptical cameras) has become more and more important. In contrast to

traditional cameras, light field cameras are able to capture both the angular and spatial information of the light rays that are propagated through space. They are thus able to obtain multiple views of the same scene in a single photographic image exposure, to estimate the depth of the scene or to shift the focus of the image after capturing the image [1]. These advantages have led to light field cameras becoming an important tool in image processing and optical metrology. As a result, a precise calibration of these cameras becomes increasingly important.

The first commercially available light field camera was presented by Ng [1]. He proposed a hand-held camera that consisted of an additional micro-lens array in front of the sensor. This array additionally allows to detect the directional dependencies of the rays, and thus a light field can be extracted. Since the design of microlens based cameras is not trivial, the light field has to be decoded from the raw sensor image using sophisticated algorithms. Furthermore, each lens (micro and main lens) is affected by the usual lens aberrations, *i. e.* a subsequent rectification of the light field is necessary to obtain correct geometric information relevant for image processing and metrology applications. Dansereau *et al.* [2] presented a method that first extracts a light field from the raw sensor data and then rectifies it by estimating the values of a 12-parameter camera model. Bok *et al.* [3], in contrast, presented a method that could extract the rectified light field directly from the raw sensor data by also using a low-dimensional camera model. In order to be able to extract any information about the light field, both methods must initially detect the microlenses very precisely. But, since the camera rays at the boundary of the microlenses are very difficult to model in both methods, these pixels are mostly discarded.

Another disadvantage of these methods is the model based calibration in general. It can't describe highly local errors such as the strong distortions at the boundaries of the microlenses using a low-dimensional model. As a consequence, in the recent years, new camera models were proposed that describe the camera as a generic imaging system. They are able to model the ray of each pixel individually and thus allow high-precision calibration [4, 5]. However, the biggest disadvantage of the common light field reconstruction methods is that they are only applicable for a single type of cam-

era, *e.g.* microlens based light field cameras whose microlenses are exactly focused on the sensor. To our knowledge, there is no single method yet that can reconstruct a light field from any type of light field camera.

In this work we present a method to reconstruct a light field, that was captured by an arbitrary light field imaging system, without knowing the actually used configuration of optical elements inside the camera. We propose to use a generic camera calibration procedure to optimally calibrate each individual pixel of the camera, where all distortions of the optical elements are contained in the unconstrained bundle of sight rays, and thus are modeled very accurately. Further, we propose to use this bundle of rays to obtain an irregularly sampled presentation of the light field, and finally, we present a simple reconstruction method to interpolate a rectified light field from the irregularly spaced camera rays. We use the presented method to calibrate and reconstruct light fields from a commercially available Lytro Illum light field camera.

The paper is organized as follows: Section 2 provides the background about light fields and light field cameras as well as an introduction to the concept of generic camera calibration. Section 3.1 and 3.2 derive the 4D light field parameters from the unconstrained ray bundle obtained in the generic calibration. Section 3.3 describes the algorithm for the reconstruction of the light field from the rays' intensity values and finally, section 4 experimentally validates the proposed method by analyzing real light field images. At last, section 5 draws conclusions and presents directions for future work.

2 Background

2.1 Light Fields and Light Field Cameras

In the field of geometrical optics the light of a scene can be described by the plenoptical function with six variables: three spatial coordinates, two angular coordinates, one spectral value. In a conventional camera usually only a subspace of this function can be captured: two spatial coordinates with a color/intensity value. A light field camera allows to capture two additional angular dimensions. For this, the most common type are microlens based light field cameras. The

design of these is similar to that of conventional cameras, with the difference that an array of microlenses is positioned in front of the sensor [1], see fig. 2.1. By adding the microlens array it is possible

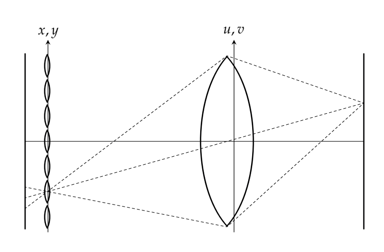


Figure 2.1: Schematic structure of the light field camera.

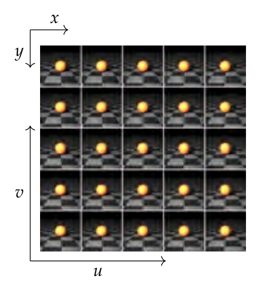


Figure 2.2: Interpretation of the light field as a camera array.

to capture a section of the light field $L(u, v, x, y)$ of a scene. Here x, y describe the coordinates of the microlenses in front of the sensor and thus, the spatial dimension of the light field. u, v describe the coordinates within the microlens relative to its center and implicitly provide information on where a light ray has passed through the main lens. They represent the angular information of the light field. Each u, v coordinate therefore represents a virtual subcamera, which observes only a part of the main lens, meaning that a light field camera can also be interpreted as a multi-camera array, whereby each subcamera has a slightly different view onto the scene, see fig. 2.2. The additional information compared to the standard camera allows to change the perspective on the scene after the exposure, which allows to extract depth information, or to shift the focus after the image capture.

In particular, there are different configurations, *e. g.*, the distance of the array to the sensor can be varied or microlenses with multiple focus lengths can be used [6]. Furthermore, there are coded aperture based light field cameras, kaleidoscope-like configurations and of course camera arrays [7, 8]. All have in common that decoding the light field from the sensor data and calibrating the camera is generally difficult. For example, to reconstruct the light field of microlens-based cameras, the centers of the microlenses, which

are often arranged in a hexagonal grid, must be detected very accurately [9]. The 4D light field can then be extracted by shifting the pixels onto a rectangular grid and reshaping the 2D-microlens-images into a 4D array. This light field, however, generally still contains all the distortions of the main lens and the microlenses, which is why an additional rectification is necessary [2,3].

2.2 Calibration

The basis of the calibration is a precise modeling of the camera, which is of course strongly influenced by the camera type. Conventionally, low-dimensional models are used to model the entire camera. However, their disadvantage is that they have insufficient descriptive power. Consequently, with modern cameras or optical systems not all pixels can be described perfectly by these few model parameters. The more complex an optical system becomes, the more difficult it is to model it using a low-dimensional representation. Hence, the lack of flexibility and precision has led to the development of new camera models. Cameras are described as generic imaging systems, which are independent of the specific camera type and allow high-precision calibration [4,5]. An imaging system is modeled as a set of photosensitive pixels, where all other optical elements are represented by a black box. Each pixel collects light from a bundle of rays entering the imaging system, which is called *raxel*. The set of all *raxels* with the associated geometric parameters forms the complete generic imaging model.

The geometric parameters can be described for each pixel i by a single camera ray running through the center of the *raxel* along the direction of light propagation, $\vec{r}_i = (\vec{d}_i^{\mathrm{T}}, \vec{m}_i^{\mathrm{T}})^{\mathrm{T}}$, with a direction vector $\vec{d}_i$ and a start vector $\vec{m}_i$. Its calibration is usually performed by minimizing the Euclidean distance of the rays $\vec{r}_i$ to known reference points $\vec{p}_{ik}$ in space, also called ray re-projection error: $\epsilon_i = \sum_k d_{\text{euclid}}(\vec{r}_i, \vec{p}_{ik})$. A minimization of the commonly used ray projection error is often not possible, because most generic models do not support a direct projection onto the pixel plane. See [5] for more details.

The advantage of this type of modeling is that there is no longer one global model that has to describe the camera over the entire pixel plane. Instead, with the generic model even high-frequency distortions in the optical imaging system can be modeled equally accurate both locally and globally, resulting in a highly accurately calibrated camera. This is specifically important for light field cameras, where it becomes very difficult to model distortions of the microlenses with a global model. In the end, however, one does not obtain an "image", but rather a set of rays with corresponding intensities. This does not interfere with many applications in optical metrology, *e. g.*, profilometry or deflectometry, where only the geometric ray properties are relevant [10]. But it can make other tasks more difficult, due to the loss of spatial correlations between pixels and their corresponding rays. The classic image processing algorithms cannot be applied without further effort. In the special case of the light field camera, algorithms such as the subsequent re-focusing of the image or a simple depth estimation can no longer be carried out using standard methods. Therefore, we propose to use the generic camera model to reconstruct the light field from the set of rays. And thus, we obtain a generic algorithm to extract the light field from an arbitrary optical imaging system, neglecting the actual design of the used light field camera.

3 Light Field Reconstruction

3.1 From Generic Camera Rays to Light Field Coordinates

In order to reconstruct the light field from the camera raw data, the camera must first be calibrated using a generic calibration method [5]. Since the camera is considered a black box, it is generally not possible to define a consistent camera coordinate system for every camera. Hence, the result of the generic calibration is not unique, *i. e.* the calibrated rays are represented in an arbitrary coordinate system, which depends on the starting configuration of the generic calibration procedure. To transform this arbitrary coordinate system into one that is fixed to the individual camera, a few steps are necessary. First, we need to define the origin as the point which has the smallest distance to all rays, *i. e.* it minimizes the mean Euclidean

distance to all rays. For a light field camera this corresponds approximately to the center of the exit pupil. Further, we define the z-axis of the camera coordinate system as the average ray direction. The last remaining degree of freedom is the rotation around this z-axis. To determine it, we calculate the intersections of all rays with a distant plane orthogonal to the z-axis. Since light field cameras project the light perspectively onto a rectangular sensor, the pattern of the intersections will be its projection into space. Applying a principal component analysis (PCA) to this 2D point cloud results in a rotation which aligns the rectangle with the x- and y-axes. As final step, we transform the rays into light field coordinates. For this, we calculate the intersections of the rays with the 2-plane-representation of the light field. The u, v-plane is placed orthogonal to the z-axis into the origin of the coordinate system. The x, y-axis is placed parallel to this at an arbitrary distance f. Thus, each ray $\vec{r}_i$ can be described by four light field coordinates (u_i, v_i, x_i, y_i) with color value $L(u_i, v_i, x_i, y_i)$, see fig. 3.1.

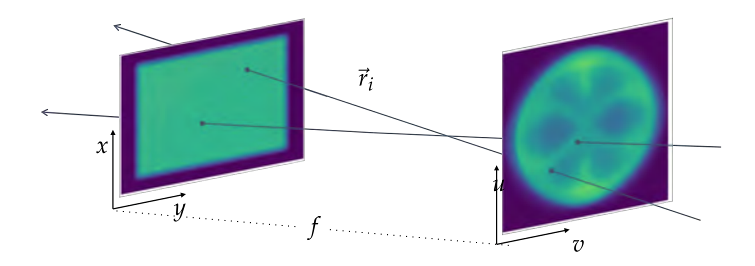


Figure 3.1: 2-plane-parameterization of the light field. The ray $\vec{r}_i$ intersects the u, v- and the x, y-plane in (u_i, v_i, x_i, y_i). The intensities in the planes visualize the spatial distribution of the intersection points as a 2D histogram.

3.2 Discrete Light Field

In order to reconstruct a light field from the bundle of rays belonging to the camera, the observed ray intensities must be interpolated to a discretized light field. We parameterize it to be interpolated into the same 2-plane-representation as before. The complete set of real camera rays described as a set of 4D-points is arranged in an

irregular 4D-grid. Still, the classical light field algorithms require a regular grid with uniform spacing. Therefore, this irregular grid of continuous rays has to be interpolated to a discrete light field described by a regular grid. The number of 4D cubes in each direction and the length of their edges could in principle be defined arbitrarily, but it is advisable to incorporate knowledge about the physical camera. For example, our microlens-based light field camera (Lytro Illum) has about 14×14 pixels under each microlens. Thus, this sampling can be used directly as a basis for the discretization of the u, v-plane. The sampling of the x, y-plane can be determined in the same way by, *e.g.*, the number of microlenses in front of the sensor. This procedure leads to a regular grid with grid points $(u, v, x, y) \in U \times V \times X \times Y$ with the resolutions of the respective dimensions $U = V = [0, \ldots, 14]$, $X = [0, \ldots, 551]$, $Y = [0, \ldots, 383]$. After the discrete target light field has been defined, we need to transform the set of real camera rays. First, by means of a histogram analysis of the spatial density of the ray-plane intersection points, the domains of the real light field dimensions are determined, see fig. 3.1. In order to place the regular grid structure into the irregular data, we define the grid extension by using a threshold value on the histogram data. A threshold of, *e.g.*, 10% ensures that most of the camera rays are within the range defined by the regular grid. Since the real light field parameters are specified in physical units, *e.g.* mm, they have to be transformed to the previously defined discrete 4D-pixel grid by a shifting and scaling operation, *e.g.* $u_i \leftarrow \frac{u_i - \min(u_i)}{\max(u_i) - \min(u_i)} U_{\max}$. This still results in irregular spaced data, which however can now be interpolated more easily to the desired regularly sampled light field.

3.3 Reconstruction

After the parameters of the light field have been defined, each corresponding light field pixel can be determined for every ray, by finding the discrete grid point that is closest to the rays' light field representation. Since the rays and the grid are normalized to the same scale, these correspondences $\mathcal{N}_{u,v,x,y}$ can easily be found by a simple rounding operation to the closest integer $[\,\cdot\,]$. As a result, each light

field pixel is only influenced by the rays that lie in the corresponding 4D-cube:

$$\mathcal{N}_{u,v,x,y} := \left\{ i \;\middle|\; 1 \geq \left\| (u,\, v,\, x,\, y)^{\mathrm{T}} - \left([u_i]\,,\, [v_i]\,,\, [x_i]\,,\, [y_i] \right)^{\mathrm{T}} \right\|_0 \right\} . \quad (3.1)$$

The intensity of a discrete pixel can then be calculated from the intensity values of the corresponding rays as a weighted average:

$$L(u,v,x,y) = \frac{1}{\sum_{i \in \mathcal{N}_{u,v,x,y}} w_i} \sum_{i \in \mathcal{N}_{u,v,x,y}} w_i \, L(u_i, v_i, x_i, y_i)\,, \quad (3.2)$$

$$w_i = \frac{1}{\epsilon_i} \exp(- \left\| (u,v,x,y)^{\mathrm{T}} - (u_i, v_i, x_i, y_i)^{\mathrm{T}} \right\|_1^2) \,. \quad (3.3)$$

For the weighting factor we calculate the distance between the ray's light field parameters and its correspondence in the grid. In order to consider larger deviations less, the error is squared and exponentially weighted. An additional weighting of the different light field coordinates is not required, since these have already been brought to a unified basis by the normalization of section 3.2. To additionally benefit from the results of the generic calibration, the error ϵ_i of the calibration procedure is taken into account, *e. g.* the pixelwise ray-projection error [5]. This suppresses badly calibrated camera rays, which often do not have good optical properties, *e. g.* dead pixels or pixels at the edges of micro lenses, which can be strongly distorted.

4 Results

For the evaluation of the proposed method, the sight rays of a Lytro Illum light field camera were estimated using a generic camera calibration. Subsequently, these were used to reconstruct the light field of a scene, using the proposed method. The reconstruction of the central view of an example image is shown in fig. 4.1. Here, only rays from the center of the u, v-plane where used in the reconstruction. For a comparison to the state-of-the-art, the methods of Dansereau *et al.* [2] and Bok *et al.* [3] were evaluated too. It can be seen that the proposed method can reconstruct the scene correctly, although there were absolutely no presumptions about the internal optical structure



Figure 4.1: Center views of the light field. Dansereau *et al.* (top left), Bok *et al.* (top right), proposed method (bottom left). Detailed views: Dansereau *et al.* (top), Bok *et al.* (middle), proposed method (bottom).

of the camera and no information of the spatially correlated pixels was used. The reconstruction results of Dansereau *et al.* and Bok *et al.* are relatively similar, but show a sharper result compared to the proposed method. In detail it can be seen that the proposed method can reconstruct the light field even near object edges very well. The visibly larger blur is due to the relatively freely chosen sampling of the light field. A better optimized choice of the light field dimensions should result in less rays being summed up, thus reducing the blur. In addition, the arbitrary offset of the reconstruction grid produces interpolation-related blur. This should also be reduced by a further optimization of this offset.

Nevertheless, the advantage of the proposed method can be found in another area. Apart from the central view, the light field contains much more information. If one fixes an angular and a spatial coordinate in the 4D light field pointing in the same direction, *e.g.* u and x, one gets a 2D-slice of the light field, a so-called epipolar plane image (EPI) [1]. Lines of different slopes can be seen, whose orientation represents the depth of the observed object point [7]. The depth

estimation is thus reduced to a simple local orientation estimation in the EPIs, whereby the quality of the estimation is significantly influenced by the calibration. The better the quality of the lines, the better the result of the depth estimation. Fig. 4.2 shows examples of a horizontal and a vertical EPI generated by fixing u and v to its center coordinates and by selecting pixel lines for the x and y coordinate, respectively. The EPI of Dansereau *et al.* shows strong deviations from the epipolar geomery, visible through the curvy epipolar lines. This is caused by the poor generalizability of the method, which was developed for the old Lytro camera and works only moderately well for the newer Lytro Illum. The EPI of Bok *et al.* on the other hand is much straighter. However, there are errors at the top and the bottom. These areas correspond to pixels which are located at the boundary of the microlenses, where the imaging is more strongly distorted. For the proposed method, it can be seen that the epipolar geometry is maintained much better, visualized by the straight lines in the EPIs. Also, the distortions of the lenses are compensated, resulting in a rectified light field. However, as before, due to generic nature of the method, the sampling is not yet ideal. This is visible by the overall lower resolution and the slightly more blurry appearance.

5 Conclusions

In this paper we presented a method that allows us to calibrate any light field camera (*e. g.* microlens-based, mirror-based, camera arrays) without having to model the exact optical properties. Using a generic calibration, we can precisely calibrate the individual camera rays. We normalized the result to transform it into an equivalent light field representation. Since classical algorithms require a regular sampling, we fit a regular 4D grid into the irregular camera rays. Summation of the rays' weighted intensity values finally resulted in the interpolation and reconstruction of the rectified light field. Experiments showed that the method can provide good reconstructions and that it returns rectified light fields. The epipolar geometry between the subcameras is preserved and shows even better results than the conventional methods. However, in detail it can be seen that the reconstructed light fields are more blurred in comparison

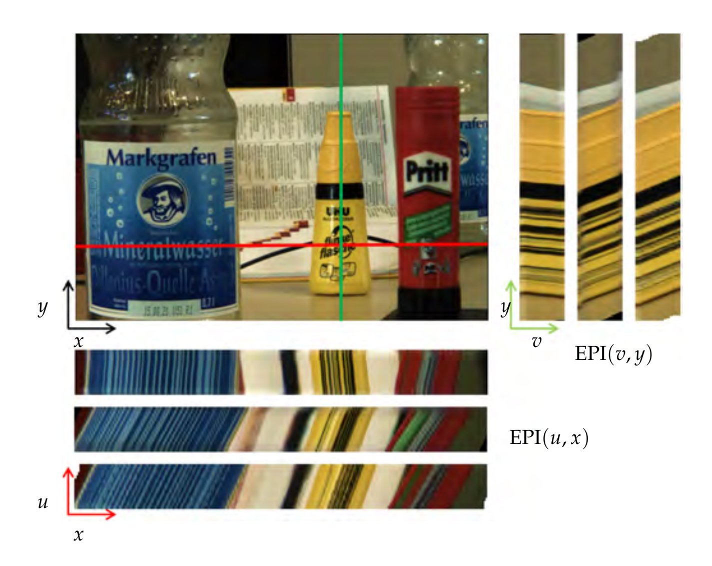


Figure 4.2: Horizontal (red) and vertical (green) EPIs in comparison: Dansereau *et al.* (top & left), Bok *et al.* (middle), proposed method (bottom & right).

to the standard methods. This can be explained by the sub optimal sampling of the light field coordinates. Therefore, further work is devoted to the improvement of the light field sampling, whereby both the desired resolution and the position of the grid points will be optimized and adapted to the used camera. Also, more experimental evaluation using different light field acquisition systems is in progress.

References

1. R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, “Light field photography with a hand-held plenoptic camera,” *Computer Science Technical Report CSTR*, vol. 2, no. 11, pp. 1–11, 2005.
2. D. G. Dansereau, O. Pizarro, and S. B. Williams, “Decoding, calibration and rectification for lenselet-based plenoptic cameras,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1027–1034.

3. Y. Bok, H.-G. Jeon, and I. S. Kweon, "Geometric calibration of microlens-based light field cameras using line features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 287–300, 2017.
4. M. D. Grossberg and S. K. Nayar, "The raxel imaging model and ray-based calibration," *International Journal of Computer Vision*, vol. 61, no. 2, pp. 119–137, 2005.
5. D. Uhlig and M. Heizmann, "A calibration method for the generalized imaging model with uncertain calibration target coordinates," in *2020 IEEE Asian Converence on Coputer Vison*, H. Ishikawa, C.-L. Liu, T. Pajdla, and J. Shi, Eds. Springer International Publishing, 2020.
6. T. Georgiev and A. Lumsdaine, Eds., *The multifocus plenoptic camera*. International Society for Optics and Photonics, 2012.
7. S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 606–619, 2014.
8. I. Ihrke, J. Restrepo, and L. Mignard-Debise, "Principles of light field imaging: Briefly revisiting 25 years of research," *IEEE Signal Processing Magazine*, vol. 33, no. 5, pp. 59–69, 2016.
9. M. Schambach and F. Puente León, "Microlens array grid estimation, light field decoding, and calibration," *IEEE transactions on computational imaging*, vol. 6, p. 591–603, 2020.
10. S. Werling, M. Mai, M. Heizmann, and J. Beyerer, "Inspection of specular and partially specular surfaces," *Metrology and Measurement Systems*, vol. 16, no. 3, pp. 415–431, 2009.

Analyse des Flug- und Abbrandverhaltens von Ersatzbrennstoffen auf Basis eines Lichtfeldkamerasystems

Miao Zhang[1], Markus Vogelbacher[1], Krasimir Aleksandrov[2], Hans-Joachim Gehrmann[2] und Jörg Matthes[1]

[1] Karlsruher Institut für Technologie, Institut für Automation und angewandte Informatik,
Herrmann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen
[2] Karlsruher Institut für Technologie, Institut für Technische Chemie,
Herrmann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen

Zusammenfassung Heutzutage finden die aus Abfällen gewonnenen Brennstoffe eine zunehmende Verwendung bei industriellen Verbrennungsprozessen, wie beispielsweise zur Erzeugung von Wärme bei der Verbrennung in Zement-Drehrohröfen. Um eine kontrollierbare und sichere Verbrennung dieses alternativen Brennstoffs zu gewährleisten, ist eine Analyse des Flug- und Verbrennungsverhaltens unerlässlich. In diesem Beitrag stellen wir Methoden zur Analyse von Bilddaten vor, die von einer Lichtfeldkamera während der Verbrennung von den aus Abfällen gewonnenen Brennstoffen in einem Drehrohr aufgenommen wurden. Das Kamerasystem liefert 3D-Informationen sowohl zu den Brennstoffpartikeln als auch zur inneren Form des Drehrohrofens. Die Analyse beinhaltet Verfahren zur Partikeldetektion unter Verwendung von 3D-Clustering-Algorithmen und Verfahren zur Partikelverfolgung unter Verwendung von Multi-Objekt-Tracking-Algorithmen.

Keywords Partikeldetektion, 3D-Clustering, Lichtfeldkamera, Multiple-Target-Tracking

1 Einleitung

Die Nutzung von Ersatzbrennstoffen (EBS) hat sich bei industriellen Verbrennungsprozessen, wie etwa der Zementherstellung, etabliert. Dabei ist neben der Kostenreduktion der große Vorteil, dass sich der biogene Anteil des EBS positiv auf die CO_2-Bilanz des Verbrennungsprozesses auswirkt. Die massenmäßig meistverwendeten EBS stellen die aufbereiteten, festen, flugfähigen Brennstoffe dar, die als FLUFF bezeichnet werden. Der FLUFF setzt sich aus einer Mischung unterschiedlicher Fraktionen, wie z. B. Papier und Pappe, Holz, Plastikfolien und 3D- Plastikpartikeln zusammen. Auf Grund der komplexen Zusammensetzung und der sich zeitlich und örtlich ändernden Partikelgrößen resultiert ein instationäres Flug- und Abbrandverhalten, was den FLUFF-Einsatz erschwert.

Um das Flug- und Verbrennungsverhalten des FLUFF besser vorhersagen und damit dessen Einsatz optimieren zu können, werden im gleichnamigen AiF-Projekt "FLUFF"3D-Verbrennungssimulationsmodelle (CFD) erarbeitet. Zur Validierung der Modelle anhand realer Messdaten werden neue kamerabasierte Verfahren zur Ermittlung der Statistik der 3D-Flugbahnen und der Zündzeitpunkte von Brennstoffpartikeln anhand von Messungen an der am Campus Nord des KIT befindlichen Versuchsanlage BRENDA entwickelt. Dazu werden die Brennstoffpartikel durch ein plenoptisches, metrisch kalibriertes Hochgeschwindigkeitskamerasystem erfasst. Darauf aufbauend werden Verfahren zur automatisches Detektion der Partikel und ein 3D-Tracking der dazugehörigen Trajektorien entwickelt.

In der Literatur sind zahlreiche Verfahren und Anwendungen zur Detektion und zum Tracking von Partikeln zu finden. Die Partikel-Detektion erfolgt meist aus den 2D-Bildinformationen beispielsweise mittels SIFT [1] oder mittels Neuronaler Netze [2]. Sind 3D-Informationen in Form von Punktwolken verfügbar (Stereokamera), kann die Detektion auch über Clustering-basierte Ansätze erfolgen [3]. Verfahren für das Brennstoffpartikel-Tracking auf Basis von 2D-Hochgeschwindigkeitskameras werden in [4, 5] vorgestellt. Ein Verfahren für das 3D-Tracking von Tracer Partikeln in Fluiden auf Basis von Stereokameras wird in [6] beschrieben. Der Einsatz

eines Lichtfeldkamerasystems zur Realisierung einer 3D-Particel-Tracking-Velocimetry (PTV) wird in [7] vorgestellt.

Aufgrund konstruktiver Randbedingungen bei Drehrohröfen und Brennkammern ist die Nutzung von Stereokamerasystemen i. d. R. nicht möglich. Daher wird in diesem Paper der Einsatz eines Lichtfeldkamerasystems für die Detektion und das Tracking von Brennstoffpartikeln untersucht. Durch die Lichtfeldkamera stehen sowohl 2D-Bildinformationen also auch 3D-Punktwolken zur Verfügung. Daher werden für den Partikel-Detektionsschritt drei Methoden verwendet: 2D SIFT, 3D DBSCAN Clustering und die Kombination von 2D- und 3D-Informationen. Das Tracking erfolgt zunächst 2D.

2 Versuchsaufbau und Bildaufnahmesystem

Der Aufbau der Anlage ist in Abbildung 2.1 dargestellt. Das enthaltene Drehrohr ist Hauptbestandteil der Versuche. Es hat eine Länge von 8.4 m und einen Innendurchmesser von 1.4 m. Über eine Lanze am Einlauf des Drehrohres können EBS-Partikel mit einem Durchmesser von 5 bis 40 mm mit Förderluftdrücken von 4 bis 5 bar eingeblasen werden. Hierbei beheizt ein ebenfalls am Einlauf befindlicher Ölbrenner das Drehrohr auf eine Innentemperatur von etwa 1240 °C. Aufgrund der hohen Temperaturen zünden die meisten EBS-Partikel auf ihrer Flugbahn durch das Drehrohr. Am Auslauf des Drehrohres kann über ein Beobachtungsfenster aus Quarzglas das Drehohrinnere z. B. über ein Kamerasystem betrachtet werden. Dabei sind neben dem heißen Drehrohr und der Ölbrennerflamme die interessierenden gezündeten Partikel und teilweise auch nicht gezündete Partikel sichtbar (Abbildung 2.2, links).

Eine Lichtfeldkamera, auch plenoptische Kamera genannt, erfasst neben den üblichen zwei Bilddimensionen noch die Tiefeninformationen. Dadurch wird eine 3D-Punktewolke (x-,y-,z-Positionen) erhalten. Im Vergleich zu einer konventionellen Kamera verfügt die Lichtfeldkamera über ein Mikrolinsen-Array (MLA) vor dem Bildsensor, wodurch die selbe Szene aus verschiedenen Blickwinkeln aufgenommen werden kann. Durch Verwendung von Mikrolinsen mit unterschiedlicher Brennweiten (multi-focus plenoptic camera) wird sowohl ein großer Tiefenschärfebereich als auch eine hohe maximale

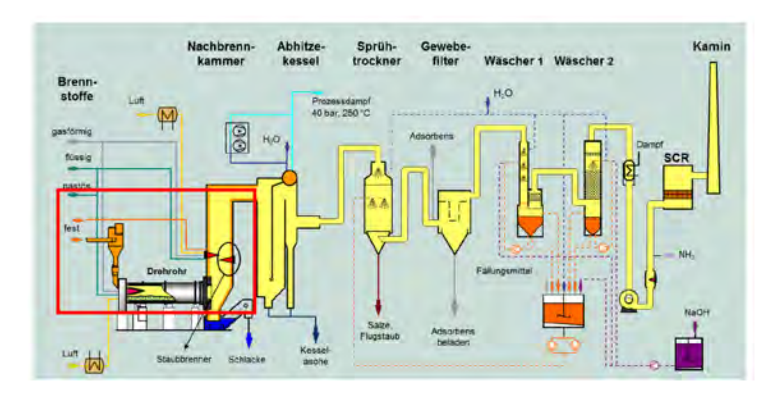


Abbildung 2.1: Aufbau der BRENDA Versuchsanlage. FLUFF Versuche werden im Drehrohr durchgeführt (rot markiert).

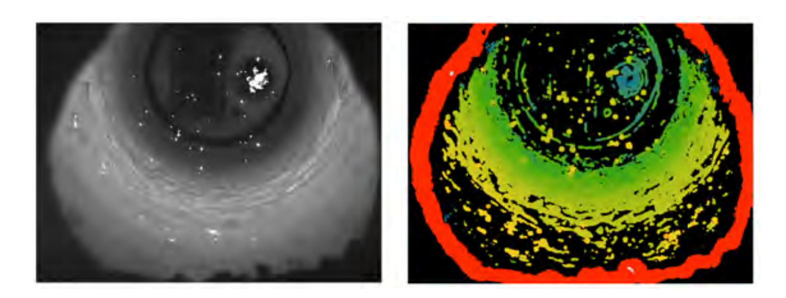


Abbildung 2.2: Beispiel einer Lichtfeldkameraaufnahme von brennenden EBS-Partikeln in einem Drehrohr. Links: Basic-Focus-Bild. Rechts: Tiefenkarte in Falschfarbendarstellung.

laterale Auflösung erreicht [8]. Durch eine vorab erfolgte Kalibrierung kann eine metrische Tiefeninformation ausgegeben werden [9]. Die eingesetzte Lichtfeldkamera der Firma Raytrix hat eine Framerate von 330 Frames pro Sekunde, eine Auflösung von 2048x1536 Pixeln und Mikrolinsen mit drei unterschiedlichen Brennweiten. Abbildung 2.2 zeigt eine Beispielaufnahme der Lichtfeldkamera unter den oben genannten Versuchsbedingungen. Zum einen das so genannte Basic-Focus-Bild (Abbildung 2.2, links), das der Aufnahme einer konventionellen 2D-Kamera entspricht, und zum anderen die

errechnete Tiefenkarte (Abbildung 2.2, rechts), die die Tiefeninformation in Falschfarben kodiert darstellt.

3 Detektion und Tracking der Brennstoffpartikel

Um das Flug- und Verbrennungsverhalten automatisch auswerten zu können, ist es zunächst notwendig, die einzelnen Brennstoffpartikel in den Lichtfeldkameraaufnahmen zu detektieren. Dabei sollen sowohl gezündete (brennende) als auch nicht gezündete bzw. ausgebrannte Partikel berücksichtigt werden. Als Datengrundlage stehen die 2D- und 3D-Informationen der Lichtfeldkamera zur Verfügung. Entsprechend können Verfahren zur Partikeldetektion in 2D und 3D genutzt werden, um eine vollständige Detektion aller Brennstoffpartikel zu erreichen.

3.1 Partikeldetektion

2D Partikeldetektion: Scale-invariant Feature Transform (SIFT)

Basierend auf dem 2D Grauwertbild der Lichtfeldkamera kann eine Partikeldetektion mittels Scale-Invariant Feature Transform (SIFT) [10] durchgeführt werden. Der Merkmalsraum des SIFT wird durch Faltung mit einem Difference of Gaussian Filter berechnet. Eine Maximasuche über die Ebenen der Difference of Gaussian Merkmalspyramide führt zu Keypoints, die in unserem Anwendungsfall als Partikeldetektionen behandelt werden. Auf Grund der Skalierungsinvarianz des SIFT können Partikel verschiedener Größen detektiert werden.

3D-Partikeldetektion: Clustering Algorithmus DBSCAN

Die 3D-Punktewolke der Lichtfeldkamera kann mit Hilfe des DBSCAN (Density-Based Spatial Clustering of Applications with Noise) Clustering Algorithmus [3] analysiert werden. Der Algorithmus definiert Kernpunkte, die innerhalb von einem bestimmten Radius ε eine Mindestanzahl von Nachbarpunkten minPts besitzen. Ein Punkt, der kein Kernpunkt ist, aber dessen Abstand zu einem Kernpunkt kleiner als ε ist, ist vom Kernpunkt dichte-erreichbar. Ein Punkt, der we-

der ein Kernpunkt noch von einem Kernpunkt dichte-erreichbar ist, wird als Rauschen definiert. Zwei Punkte, die durch eine Kette von Kernobjekten, die untereinander dichte-erreichbar sind, miteinander verbunden werden können, gelten als dichte-verbunden und bilden mit Punkten, die mit denselben Kernpunkten dichte-verbunden werden können, ein Cluster. Zum Cluster werden auch die zur Verbindung benötigten dichte-erreichbaren Kernpunkte gezählt. Dichte-erreichbare Punkte, die von mehr als einem Cluster dichte-erreichbar sind, werden zufällig dem ersten möglichen Cluster zugeordnet.

Kombination von SIFT und DBSCAN Clustering

Die beiden zuvor vorgestellten Verfahren der SIFT Partikeldetektion und des DBSCAN Clustering Algorithmus werden im Folgenden für ein besseres Detektionsergebnis miteinander kombiniert. Dabei wird ausgenutzt, dass beide Verfahren unterschiedliche Informationen der Lichtfeldkamera nutzen. Partikel mit niedriger Helligkeit werden z. B. durch das grauwertbasierte SIFT Verfahren nicht erkannt, sind aber durch das auf geometrische Zusammenhänge achtende DBSCAN Clustering detektierbar. Im Gegenzug sorgen dunkle hervorstehende Kanten des Drehrohres beim Clustering für Cluster, die keine Partikel und damit falsche Detektionen darstellen. Diese werden beim SIFT nicht berücksichtigt und können durch eine entsprechende Kombination mit dem Clustering herausgerechnet werden. Die zur Kombination von 2D-SIFT und 3D-Clustering notwendige Umrechnung von 2D- in 3D-Koordinaten und umgekehrt, ist durch eine Lookup Tabelle von der Kamera gegeben.

Abbildung 3.1 zeigt eine schematische Darstellung der Vorgehensweise der aus SIFT und Clustering kombinierten Partikeldetektion. Im ersten Schritt werden mögliche Problemfälle beim Clustering identifiziert. Dies ist zum einen die Drehrohrwand, die fehlerhafte Cluster erzeugen kann bzw. an der auf Grund der in diesem Bereich dicht liegenden 3D-Informationen immer ein großes Cluster entsteht, und zum anderen die Ölbrennerflamme am Einlauf des Drehrohres, die nicht zur Auswertung herangezogen werden soll. Zur Detektion der Drehrohrwand wird der Clustering Algorithmus mit vergleichsweise großem ε=50 mm und kleinem minPts=6 auf die komplette 3D-Punktewolke angewandt. Das größte erkannte Cluster wird als

Drehrohrwand gewählt und im Folgenden aus der 3D-Punktewolke entfernt. Der Flammenbereich kann auf Grund seiner hohen Helligkeit durch Anwendung des Otsu-Schwellwertverfahrens auf das Grauwertbild segmentiert werden und nach Umrechnung in 3D-Koordinaten ebenfalls aus der 3D-Punktewolke entfernt werden. Nach Entfernung der möglichen Fehlerquellen wird das Clustering auf die bereinigte 3D-Punktewolke mit einem vergleichsweise kleinem ε=15 mm und einen großem minPts=10 zur Partikeldetektion angewandt. Alle Punkte eines detektierten Clusters bilden ein Partikel, dadurch ist neben der 3D-Position auch die Geometrie des Partikels näherungsweise bekannt.

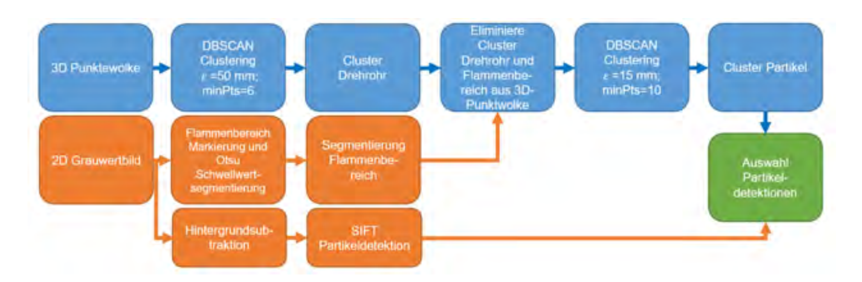


Abbildung 3.1: Schematische Darstellung des Ablaufes der kombinierten Partikeldetektion

Vor der Anwendung des SIFT auf das Grauwertbild wird zunächst eine Hintergrundsubtraktion durchgeführt. Hierzu wird vom aktuellen Grauwertbild ein über mehrere Bilder zeitlich gemitteltes Grauwertbild abgezogen, dadurch werden konstante Strukturen der Umgebung und vor allem der Drehrohrwand entfernt. Die anschließend vom SIFT gelieferten Keypoints werden als Partikelpositionen übernommen und liefern damit die 2D-Koordinaten der Partikelschwerpunkte.

Für die Kombination beider Verfahren werden die Ergebnisse aus dem Clustering in 2D-Koordinaten transformiert. Anschließend wird überprüft, ob und wie viele Keypoints der SIFT Partikeldetektion innerhalb eines detektierten Clusters liegen. Beim Vergleich können insgesamt vier Fälle eintreten:

1. Ein einzelner Keypoint liegt innerhalb eines Clusters.

2. Mehrere Keypoints liegen innerhalb eines Clusters.
3. Kein Keypoint liegt in einem Cluster.
4. Es existiert ein Keypoint, der keinem Cluster zugeordnet werden kann.

Für den Fall 1, dass nur ein Keypoint in einem Cluster liegt, wird dieses Cluster direkt als Partikeldetektion übernommen. Fall 2 mit mehreren Keypoints innerhalb eines Clusters kann verschiedene Gründe haben. Liegen mehrere Partikel räumlich nah nebeneinander, werden diese beim Clustering zu einem großen Partikel zusammengefasst, während das SIFT mehrere Partikeldetektionen liefert. Außerdem treten bei großen Partikeln, die ein Cluster darstellen, beim SIFT meistens mehrere Keypoints auf. Um unterscheiden zu können, ob in diesen Fällen ein oder mehrere Partikel vorliegen, wird der Grauwertverlauf innerhalb des Clusters betrachtet. Sind mehrere lokale Intensitätsmaxima vorhanden wird das Cluster in mehrere Partikel aufgeteilt. Ist dies nicht der Fall wird das Cluster als ein Partikel übernommen. Für Fall 3 und 4, in denen nur eines der beiden Verfahren ein Partikel detektiert hat, werden ebenfalls die Grauwerte innerhalb des Clusters bzw. in der Umgebung des Keypoints herangezogen. Durch Abgleich des Grauwertverlaufes mit einer Gauß-Verteilung wird entschieden, ob es sich tatsächlich um ein Partikel oder nur um eine Fehldetektion (z. B. durch Rauschen) handelt.

3.2 Partikel-Tracking

Für die detektierten Brennstoffpartikel wird ein Tracking durchgeführt. Für die Aufgabenstellung des Multiple-Target-Tracking (MTT) wird der in der Literatur häufig verwendete Global Nearest Neighbor (GNN) Algorithmus verwendet [11]. Der GNN enthält die Schritte Prediction, Gating, Assignment und Update. Für die Prädiktion und das Update der Position eines Partikels (Tracks) wird ein lineares Kalman-Filter verwendet. Vereinfacht wird dabei für jeden Zeitschritt eine gleichförmige Bewegung eines Partikels mit konstanter Geschwindigkeit angenommen. Bei einer großen Anzahl an Partikeln im MTT, verfügen die meisten Tracks über mehr als eine Messung im Gating-Bereich bzw. eine Messung liegt im

Gating-Bereich mehrerer Tracks. Zur Lösung des Zuordnungsproblems wird der Kuhn-Munkres-Algorithmus genutzt. Durch Minimierung einer Kostenmatrix, die den Mahalanobis-Abstand zwischen allen möglichen Messungen und Tracks berücksichtigt, wird die optimale Zuordnung zwischen Messungen und Tracks durchgeführt. Dabei werden auch vorgebbare Kosten für nicht fortgeführte Tracks berücksichtigt.

4 Ergebnisse

4.1 Ergebnis der Partikeldetektion

Abbildung 4.1 zeigt ein Beispiel der 2D- (Basic-Focus, Grauwertbild) und 3D-Kamerinformation (3D-Punktewolke), auf dessen Basis die Partikeldetektion durchgeführt wird. Zur Bewertung der Verfahren steht eine manuell gelabelte Ground Truth zur Verfügung, die für das ausgewählte Bild 126 Partikel enthält. Wie in Abschnitt 3.1 erläutert, werden vor dem Clustering zur Partikeldetektion die Drehrohrwand und der Flammenbereich der Ölbrennerflamme detektiert. Der Flammenbereich wird mit Hilfe des Otsu-Schwellwertverfahrens innerhalb des in Abbildung 4.1(a) rot markierten Rechtecks segmentiert. Durch einen ersten Clustering-Prozess kann aus der 3D-Punktewolke in Abbildung 4.1(b) ein Cluster für die Drehrohrwand gewonnen werden.

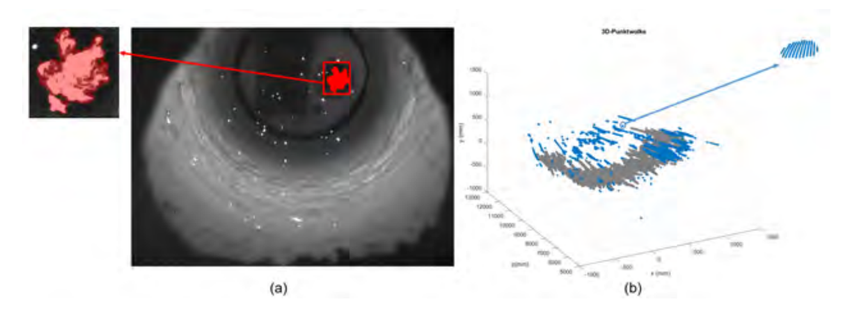


Abbildung 4.1: Datengrundlage der Lichtfeldkamera. (a) Basic-Focus-Bild (Grauwertbild) und (b) die entsprechende 3D-Punktewolke.

Punkte der Drehrohrwand und des Flammenbereichs werden aus der 3D Punktewolke entfernt. Insgesamt werden 255 Cluster erkannt, wobei nur 88 Cluster korrekt detektierte Partikel darstellen. Das SIFT Verfahren liefert 118 Keypoints von denen 95 korrekt detektierten Partikeln entsprechen. Die Kombination von Clustering und SIFT führt zu einer Detektion von 120 Partikeln, davon 116 korrekt detektierte Partikel. Tabelle 1 fasst die Ergebnisse der einzelnen Verfahren noch einmal zusammen.

Tabelle 1: Vergleich zwischen DBSCAN Clustering, SIFT Partikeldetektion und deren Kombination.

Methode	Anzahl detektierter Partikel	Anzahl korrekt detektierter Partikel	Anzahl von Fehldetektionen	Recall	Precision	F-score
DBSCAN	255	88	167	69.84%	34.51%	0.4619
SIFT	118	95	23	75.40%	80.51%	0.7787
DBSCAN+SIFT	120	116	4	92.06%	96.67%	0.9431

4.2 Ergebnis des Partikel-Trackings

Zur Beurteilung des Trackingverfahrens wird eine qualitative Auswertung anhand einer Beispielaufnahme durchgeführt. Aufgrund großer Ungenauigkeit in der Tiefeninformation der aktuell vorliegenden Messdaten wird das Partikel-Tracking zunächst in 2D durchgeführt. Grundlage bilden die Partikeldetektionen aus dem kombinierten Clustering/SIFT Verfahren für alle Bilder der Beispielaufnahme. Die Beispielaufnahme enthält 50 Bilder. Das entspricht bei einer Framerate der Kamera von 330 fps einem Zeitraum von 0.149 s. Das Ergebnis des GNN Algorithmus ist in Abbildung 4.2 dargestellt. Partikel aus einem Zeitschritt sind mit der gleichen Farbe markiert. Zugeordnete Partikel aus aufeinander folgenden Zeitschritten sind mit Pfeilen verbunden.

Mit Hilfe des Tracking Verfahrens wird eine Verfolgung der meisten Partikel über die komplette Beispielaufnahme ermöglicht. Probleme entstehen bei der Zuordnung von sehr dicht liegenden Partikeln.

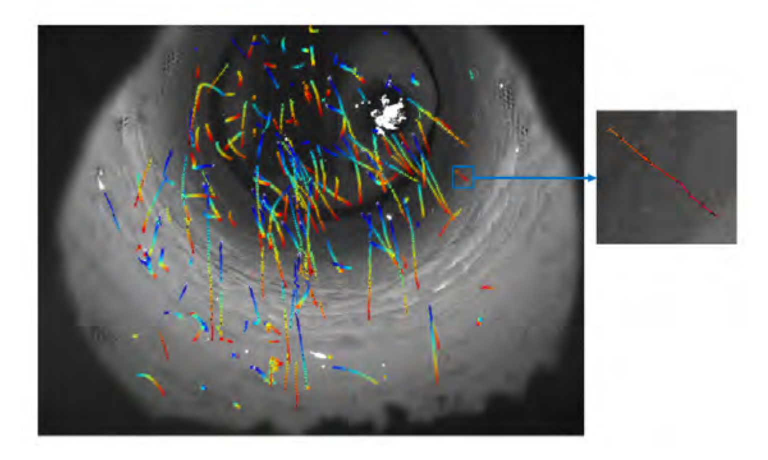


Abbildung 4.2: Ergebnis der Partikeldetektion mittels Kombination von DBSCAN Clustering und SIFT Partikeldetektion.

5 Zusammenfassung und Ausblick

Die Anwendung einer Lichtfeldkamera in einer Drehrohrumgebung ermöglicht es, die Bewegungen von im Drehrohr fliegenden EBS-Partikeln zu beobachten und detailliert zu beschreiben. Im vorliegenden Beitrag werden die Schritte Partikeldetektion und Partikel-Tracking vorgestellt, die für eine Analyse des Flugverhaltens einzelner Partikel notwendig sind. Zur Partikeldetektion wird hierzu sowohl die 2D- als auch 3D-Information der Lichtfeldkamera genutzt. Durch Kombination von 2D-SIFT und 3D-DBSCAN-Clustering wird eine effektive Detektion der EBS-Partikel erreicht. Probleme der einzelnen Verfahren, wie das Nichterkennen kleiner, dunkler Partikel durch das SIFT oder Fehldetektionen an der Drehrohrwand beim Clustering Verfahren, werden durch die Kombination beider Verfahren gelöst. Im Anschluss wird basierend auf den Partikeldetektionen durch einen GNN Algorithmus unter Nutzung eines Kalman-Filters für jeden Partikel ein Tracking durchgeführt. Das Flugverhalten der Partikel kann anhand der Partikel-Tracks analysiert werden. Durch zusätzliche Beobachtung des Helligkeitsverlaufes eines Tracks ent-

lang seiner Trajektorie kann außerdem das Abbrandverhalten zeitlich und örtlich beurteilt werden.

In folgenden Arbeiten wird untersucht, inwieweit andere Tracking-Verfahren, wie etwa Probabilistic Data Association Filter (PDAF) oder Joint Probabilistic Data Association Filter (JPDAF) die Schwierigkeiten des GNN Algorithmus vor allem bei dicht liegenden Partikeln beheben können. Des Weiteren werden zusammen mit dem Kamerahersteller Versuche zur Reduktion der Ungenauigkeit der Tiefeninformation der Lichtfeldkamera durchgeführt, um ein korrektes Tracking der Partikelflugbahnen auch in 3D-Koordinaten umzusetzen.

Literatur

1. D.-G. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision. Band 60, Nr. 2, pp. 91–110*, 2004.
2. J. M. Newby, A. M. Schaefer, P. T. Lee, M. G. Forest, and S. K. Lai, "Convolutional neural networks automate detection for tracking of submicron-scale particles in 2d and 3d," *Proceedings of the National Academy of Sciences*, vol. 115, no. 36, pp. 9026–9031, 2018. [Online]. Available: https://www.pnas.org/content/115/36/9026
3. M. Ester, J. Sander, H.-P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," in *ACM Transactions on Database Systems, Vol. 42, No. 3, Article 19*, July 2017.
4. J. Matthes, J. Hock, P. Waibel, A. Scherrmann, H.-J. Gehrmann, and H. Keller, "A high-speed camera based approach for the on-line analysis of particles in multi-fuel burner flames," in *Experimental Thermal and Fluid Science 73 (2016), pp. 10-17*, 2016.
5. Y. Xuan, J. Pei, and Y. Wanhai, "Firing particle flow detection and tracking in sequence images," in *Proceedings of the 3rd World Congress on Intelligent Control and Automation, 2000, pp. 2666–2670*, 2000.
6. Y.-G. Guezennec, R.-S. Brodkey, N. Trigui, and J.-C. Kent, "Algorithmsfor fully automated three-dimensional particle tracking velocimetry," in *Experiments in Fluids17(1994), no. 4, pp. 209–219*, 1994.
7. K. Ohmi, S. Tuladhar, and J. Hao, "Light field camera based particle tracking velocimetry," in *18th International Symposium on the Application of Laser and Imaging Techniques to Fluid Mechanics*, Lisbon, Portugal, July 2016.

8. C. Perwaß and L. Wietzke, “Single lens 3d-camera with extended depth-of-field.”
9. C. Heinze, S. Spyropoulos, S. Hussmann, and L. Wietzke, “Automated robust metric calibration algorithm for 3d camera systems.”
10. D.-G. Lowe, “Object recognition from local scale-invariant features,” in *ICCV ’99 Proceedings of the International Conference on Computer Vision. Band 2, pp. 1150–1157*, 1999.
11. P. Konstantinova, A. Udvarev, and T. Semerdjiev, “A study of a target tracking algorithm using global nearest neighborapproach,” in *International Conference on Computer Systems and Technologies - CompSysTech’2003*, 2003.

Ein Portal zur interaktiven geometrischen Inspektion großer mechanischer Bauteile

A Portal for Interactive Geometry Inspection of Large Mechanical Parts

Steffen Sauer[1], Michael Heizmann[2] und Dirk Berndt[1]

[1] Fraunhofer Institut für Fabrikbetrieb und -automatisierung IFF, Magdeburg
{steffen.sauer, dirk.berndt}@iff.fraunhofer.de
[2] Karlsruher Institut für Technologie (KIT)
michael.heizmann@kit.edu

Zusammenfassung This paper presents a novel approach for interactive inspection of large mechanical parts. Therefore we use a linear moveable portal, which is equipped with a multi sensor head that consists of a fringe projection sensor and an Augmented Reality camera system. The portal covers a measuring volume of 8.0 x 3.0 x 0.8 m^3 and uses an optical Motion Capturing System to track the sensors position and orientation in a global reference frame. Inspection preparation and execution is assisted by sensor data simulation. We show that using Augmented Reality a user can easily detect rough geometry deviations. For detailed quality inspection a user can acquire 3D point clouds which are evaluated automatically.

Keywords Quality inspection, tracking, 3D measurement, augmented reality, simulation

1 Problemstellung und Motivation

Bei der CNC-Fertigung von Werkstücken werden mit Hilfe von Bearbeitungsmaschinen Rohteile zerspanend bearbeitet. Als Rohteile kommen einfache Halbzeuge aber auch komplexere Guss- und Schmiedeteile sowie Schweißkonstruktionen zum Einsatz. Dabei ist

es von enormer Wichtigkeit, dass die zu bearbeitenden Werkstücke der vorgegebenen Rohteilgeometrie entsprechen und eine zulässige Maßtoleranz nicht überschreiten. Bei geometrischen Abweichungen außerhalb der Toleranzen kann es ansonsten zu ungewollten Kollisionen zwischen Bearbeitungswerkzeug und dem Rohteil kommen, die im schlimmsten Fall zu einem Totalausfall der Maschine führen können. Die zulässigen Toleranzen sind anwendungsabhängig, liegen bei Rohteilen, die mit Portal-CNC-Maschinen bearbeitet werden, jedoch im unteren einstelligen Millimeterbereich. Um derartige Schäden zu vermeiden, stellt die geometrische Inspektion der Rohteilen daher probates Mittel dar,.

Neben taktilen Vermessungen wird in [1] beschrieben, dass eine berührungslose 3D-Vermessung der Rohteile in der Maschine bei der Ausrichtung und Erkennung von Formabweichungen hilfreich sein kann. Hierbei bleibt zu beachten, dass sowohl für die Inspektion als auch für die Behebung von Fehlern wertvolle Bearbeitungszeit verloren geht, wenn das Rohteil bereits in der Maschine lieht. Dies führt im Allgemeinen zu wenig Akzeptanz beim Anwender. Für große Bauteile und Baugruppen existieren Lösungen, die vollautomatisiert einen Soll-Ist-Abgleich [2]. Diese fahren meist feste Prüfprogramme ab und sind wenig flexibel bei Bauteilen mit sehr geringen Losgrößen und starken unvorhergesehenen Geometrieabweichungen. Handgeführte 3D-Sensoren sind hingegen für Bauteile mit geringen Losgrößen besser geeignet. In [3] wurde das Genauigkeitspotenzial derartiger Scanner untersucht und gezeigt, dass sie Absolutgenauigkeiten bieten, die für diese Aufgabe ausreichend sind. Allerdings referenzieren sich diese Sensoren durch zusätzlich aufzubringende Merkmale am Objekt und erfassen nur scannend, was ihr Einsatzgebiet einschränkt.

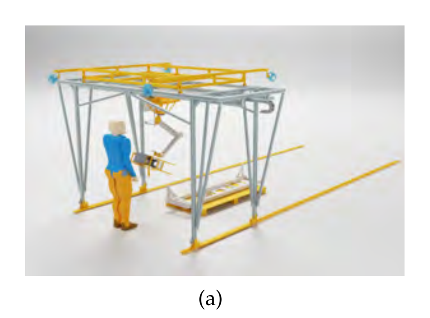
2 Lösungsvorschlag

Zur Inspektion großer Rohteile wurde daher ein Inspektionssystem entwickelt, welches die zuvor beschriebenen Probleme versucht zu beheben. Ziel der Entwicklung war es, das System so zu gestalten, dass es außerhalb der Bearbeitungsmaschine für große Rohteile eingesetzt werden kann, durch interaktive Bedienung für kleine

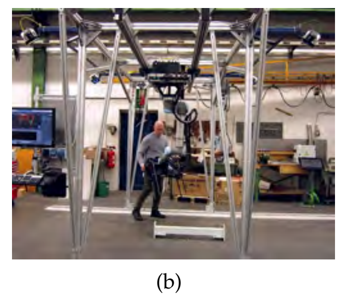
Losgrößen geeignet ist und Toleranzabweichungen im Bereich von $\pm 2,0$ mm erkannt werden können. Die Soll-Geometrie wird dabei durch das CAD-Modell der Rohteile bestimmt. Der Aufwand für die technische Realisierung soll deutlich unter dem für Koordinatenmessmaschinen liegen.

2.1 Portalaufbau

Zur partiellen dreidimensionalen Erfassung der Rohteile wurde ein flächig messender Sensor *SurfaceCONTROL* eingesetzt, der auf dem Phasenshift-Verfahren beruht [4] und in einem Arbeitsabstand von 800 mm eine ca. A3 große Fläche erfasst. Über eine Mehrachs-Kinematik, die auf Knopfdruck arretiert werden kann, ist der Sensor kopfüber mit einem Portal verbunden. Unterhalb dieses Portals kann der Sensor in einem Volumen von 2,0 x 3,0 x 0,8 m^3 manuell frei bewegt werden. Das Portal selbst ist auf Schienen gelagert und kann auf diesen ebenfalls manuell verschoben werden (s. Abb. 2.1). Damit vergrößert sich das gesamte Messvolumen auf 8,0 x 3,0 x 0,8 m^3.



(a)



(b)

Abbildung 2.1: Protoyp des Inspektionsportals: (a) CAD-Modell und (b) Bediener bei der Inspektion eines Bauteils am umgesetzten Aufbau.

Der Sensor wurde um eine weitere CCD-Kamera ergänzt, die fest in das vorhandene Gehäuse integriert wurde. Sie ist in Sichtrichtung des Sensors ausgerichtet, und besitzt einen horizontalen Öffnungswinkel von 35°. Auf der Rückseite des Sensors wurde ein 12″- Touchscreen montiert, über den die gesamte Interaktion mit dem System läuft.

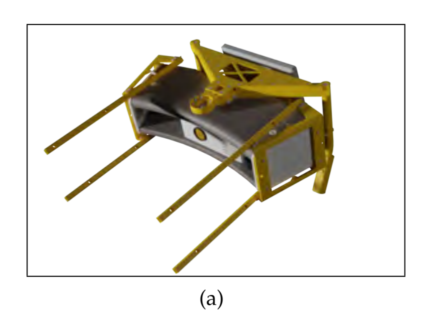
Das Kamerabild wird live auf dem Display ausgegeben und es wird kontinuierlich und lagekorrekt das CAD-Modell des zu prüfenden Rohteils als Soll-Geometrie eingeblendet. Es entsteht somit eine Augmented-Reality- (AR)-Anwendung mit der ein Bediener bereits grobe Abweichungen intuitiv erkennen kann. Mit Hilfe des 3D-Sensors können dann selektiv Messdaten von dem zu prüfenden Rohteil aufgenommen werden. Diese werden dann in das Koordinatensystem des Rohteil-CAD-Modells transformiert und die Abweichung als Falschfarbendarstellung visualisiert.

2.2 Transformation von Messdaten

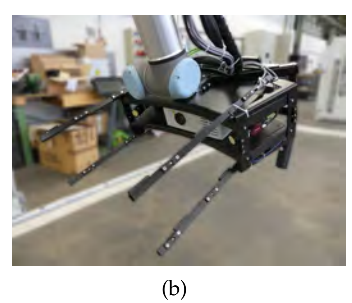
Um Messdaten, die in einem Sensorkoordinatensystem entstehen, mit einem CAD-Modell in Verbindung zu bringen, ist es erforderlich sie mittels einer Transformation $T_{SC} : Sensor \rightarrow CAD$ in das CAD-Koordinatensystem zu transformieren. Die Bestimmung dieser Transformation erfolgt in der hier vorgestellten Lösung über ein externes Tracking-System.

In [5] und [6] wurden kamerabasierte Motion Capturing Systeme, die ursprünglich für die Erfassung von menschlichen Bewegungen entwickelt wurden, hinsichtlich ihrer 3D-Punkt-Genauigkeiten analysiert. Abhängig von Aufbau und Messvolumen konnten Absolutgenauigkeiten von $<0,1$ mm nachgewiesen werden. Diese Tracking-Systeme zählen typischerweise zu den Outside-In-Tracking-Systemen, d.h. ein von außen beobachtendes System schätzt Lage und ggf. Orientierung eines beobachteten Körpers. Sie arbeiten Marker-basiert, wobei retroreflektierende passive Kugeln oder zur Steigerung der Robustheit aktiv leuchtende LEDs zum Einsatz kommen. Die Marker werden in Kamerabildern gefunden und es wird durch Vorwärtsschnitt aus mehreren Kameraperspektiven die zugehörige 3D-Koordinate berechnet. Zur Berechnung einer 6D-Transformation können mehrere Marker, die an einem Körper, genant *Body*, befestigt sind, zu einander eingemessen werden. Das Tracking-System versucht dann, in alle gefundenen 3D-Koordinaten bekannte *Bodies* durch die Methode der kleinsten Fehlerquadrate einzupassen. Das Ergebnis ist dann eine 6D-Transformation von *Body*- zu Tracking-Koordinatensystem. In der hier vorgestellten Lösung wurde ein Tracking-System der Firma OptiTrack mit vier Kame-

ras verwendet, die im oberen Bereich des Portals angeordnet und fest mit diesem verbunden sind. Sie decken den Bewegungsraum des Sensors vollständig ab. Zum Tracken des Sensors wurde dieser zusätzlich mit aktiven Marken in Form von Infrarot-LEDS versehen, die an einem Exoskelett um den Sensor herum angebracht sind (s. Abb. 2.2). Die Tracking-Software bietet eine externe Schnittstelle und liefert mit bis zu 180 Hz die Transformation T_{BO} : *Body* $\rightarrow$ *OptiTrack*.



(a)



(b)

Abbildung 2.2: Modifizierter Streifenlichtsensor: (a) CAD-Modell mit zusätzlicher Augmented-Reality-Kamera, Tracking-Skelett, Griffen und Touchscreen. (b) die Umsetzung am Inspektionsportal.

Die Messdatenerzeugung erfolgt sowohl durch einen ein 3D-Sensor als auch durch eine 2D-Kamera. Zur Vereinfachung wird im Folgenden der allgemeine Begriff *Sensor* verwendet, der sich je nach Kontext auf eines der beiden Geräte bezieht. Die Orientierung des Sensors zum *Body* kann durch eine Hand-Auge-Kalibrierung bestimmt werden. Bei unserer Lösung wurde die numerische Auswertung mit den in [7] vorgestellten Verfahren durchgeführt. Das Ergebnis ist die Transformation T_{SB} : *Sensor* $\rightarrow$ *Body*, die das Sensorkoordinatensystem in das *Body*-Koordinatensystem überführt.

Mit den beschrieben Transformationen kann somit die Lage des Sensors unterhalb des Portals berechnet werden. Um die Verschiebung des Portals zu berücksichtigen wurde das Tracking-System erweitert, indem in einem eindeutig codierten Abstandsmuster weitere Infrarot-LEDs in den Fußboden eingelassen wurden. Diese LEDs werden ebenfalls vom Tracking-System erfasst und die Ebene des Fußbodens wird damit ebenfalls zu einem *Body*. Bei einer Verschie-

bung des Portals auf den horizontalen Schienen kommt es zu einer Relativbewegung, die das Trackingsystem als zusätzliche *Body*-Bewegung interpretiert. Die inverse Anwendung dieser Transformation ergibt somit die absolute Lage des Portals bezüglich der Schienen. Da in diesem Fall das Tracking-System seine Umwelt beobachtet, handelt es sich bei diesem Verfahren um ein *Inside-Out*-Tracking-System, da das System seien eigene Lage an externen Referenzmerkmalen bestimmt. Die Marken im Fußboden wurden mit einem Laser-Tracker eingemessen und liegen im als Welt definierten Koordinatensystem. Das Tracken des Fußbodens ergibt die Transformation $T_{WO} : Welt \rightarrow OptiTrack$.

Zu inspizierende Rohteile können frei innerhalb des Messvolumens abgelegt werden. Die Transformation des Bauteils zum Weltkoordinatensystem kann ermittelt werden, indem das Rohteil an vorgegebenen Flächen mit dem 3D-Sensor erfasst wird und die resultierenden 3D-Punkte durch Abstandsminimierung in das CAD-Modell eingepasst werden. Diese Punktwolken-zu-CAD-Registrierung ergibt die Transformation $T_{CW} : CAD \rightarrow Welt$.

Die gesamte Transformationskette des Systems ist in Abb. 2.3 dargestellt. Es zeigt sich, dass die 6D-Sensor-Transformation zum CAD-Modell T_{SC} durch Hintereinanderausführung der beschriebenen Transformationen berechnet werden kann:

$$T_{SC} = T_{CW} \times T_{WO}^{-1} \times T_{BO} \times T_{SB} \tag{2.1}$$

2.3 Messdatenvisualisierung

Bei der Ausführung der Inspektion kann der Benutzer den Sensor frei im Messvolumen bewegen. Dabei kann wahlweise die AR-Darstellung zur subjektiven Bewertung angewählt werden oder es können 3D-Messdaten aufgenommen und ausgewertet werden. Während der Bewegung wird der Sensor kontinuierlich vom Tracking-System erfasst und durch die zuvor beschriebene Koordinatentransformation wird die Lage der AR-Kamera im CAD-Koordinatensystem berechnet. Durch Anwenden der intrinsichen Kameraparameter auf eine virtuelle Kamera und Übertragen ihrer Lage und Orientierung, können synthetische Bilder des zu inspi-

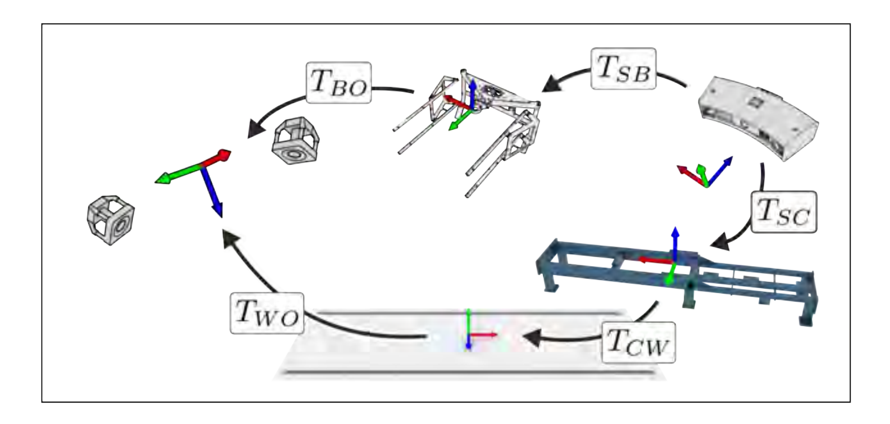


Abbildung 2.3: Koordinatentransformationen innerhalb des Inspektionsportals, um die gesuchte Transformation T_SC zu berechnen.

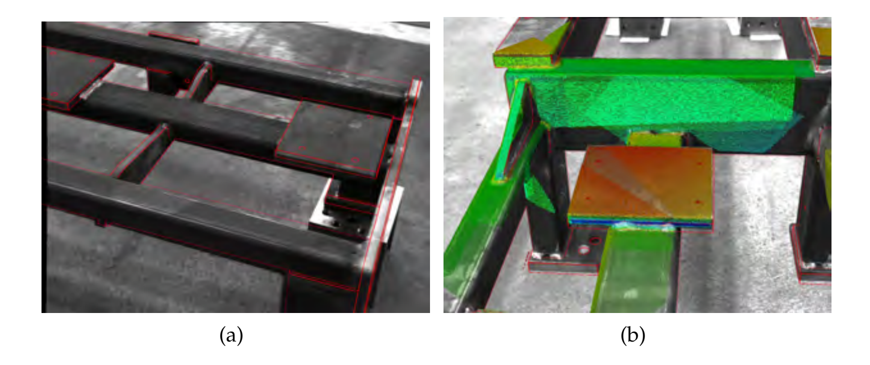
zierenden Bauteils generiert werden, die den Soll-Zustand darstellen. Durch Überlagerung des Sollzustands auf das reale Kamerabild entsteht dann eine AR-Darstellung, die eine qualitative Bewertung durch den Anwender ermöglicht (s. Abb. 10.4(a)).

Mit dem 3D-Sensor aufgenommene Messpunkte im Sensorkoordinatensystem x_S können unter Anwendung von Gl. 2.1 in das CAD-Koordinatensystem transformiert werden:

$$x_C = T_{SC} \times x_S \tag{2.2}$$

Eine gängige Methode in der geometrischen Qualitätsprüfung besteht dann darin, die Abstände der gemessenen Punkte zur Oberfläche der Soll-Geometrie zu bestimmen und in einer Falschfarbendarstellzung zu visualisieren. Durch optionales Einblenden der farbcodierten 3D-Messdaten in das Kamerabild erhält der Anwender bereits einen Überblick über die schon erfassten Bauteiloberflächen und deren Abweichung (s. Abb. 10.4(b)).

Der 3D-Sensor benötigt nach Auslösen der Messung ca. 2 s, um die erforderlichen Muster zu projizieren und die Messpunkte zu rekonstruieren. Für den Anwender ist die Ausrichtung des Sensors unter Umständen nicht trivial, da er frei im Raum positionieren werden kann. So kommt es schnell zu leeren Messdaten, wenn das Objekt



(a) (b)

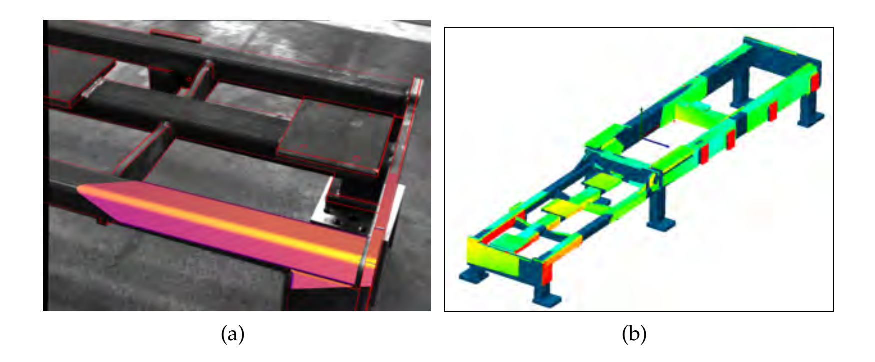
Abbildung 2.4: Ergebnisvisualisierung für den Anwender: (a) Live-Augmented Reality. Man beachte die horizontale Abweichung, die das Bauteil besitzt, (b) zusätzliche Einblendung der metrischen Abweichung durch Falschfarbencodierung.

nicht im Messvolumen des Sensors liegt. In [8] wurde bereits ein Verfahren vorgestellt, mit dem 3D-Messdaten von optischen Sensoren simuliert werden können. Diese Simulation wurde ebenfalls in die Benutzerschnittstelle integriert und kann optional aktiviert werden (s. Abb. 10.5(a)). Sie visualisiert live, welche Messdaten bei der aktuellen Sensorausrichtung im Optimalfall entstehen würden. Dies trägt zur Entlastung des Anwenders bei, da so fehlerhafte Ausrichtungen weitestgehend vermieden werden.

Zur abschließenden Bewertung können alle aufgenommenen Messdaten in Falschfarbencodierung im Kontext des CAD-Modells frei betrachtet werden (s. Abb. 10.5(b)).

3 Ergebnisse

Die Ergebnisevaluation fokussiert sich auf die Messgenauigkeit des Gesamtportals. Eine quantitative Bewertung erfolgt anhand der 3D-Messdaten, da diese metrisch gemessen werden. Die Untersuchungen sind angelehnt an die VDI/VDE-Norm 2617 [9], mit der sich die Güte 3-dimensionaler Messgeräte mit zusätzlichen Achsen objektiv bewerten lässt.



(a) (b)

Abbildung 2.5: Unterstützung bei Aufnahme und Auswertung (a) Kontinuierliche Darstellung der Messdatensimulation im Kamerabild, (b) Gesamtdarstellung aller Abweichungen von erfassten Oberflächen im Kontext des CAD-Modells.

Im ersten Versuch wurden die *Antastabweichung* des Systems bestimmt. Dazu wurde eine Kalibrierkugel mit einem Radius von 50 mm an 19 Positionen im gesamten Messvolumen verteilt und jeweils aus 3 unterschiedlichen Richtungen mit dem 3D-Sensor erfasst. Pro Kugelposition entstanden zwischen 55.000 und 97.000 Einzelmesspunkte. In die Messdaten wurde durch Abstandsminimierung jeweils eine Ausgleichskugel mit freiem Radius eingepasst, wobei maximal 3% der Messpunkte verworfen wurden. Die Antastabweichung PF ist dann die Differenz zwischen maximalen und minimalem Messpunktabstand zum jeweiligen Kugelzentrum. Für das Portal wurde nach dieser Methode ein maximaler Wert von PF =2,88 mm ermittelt. Das Histogramm über alle gemessenen Abweichungen ist in Abb. 10.1(a) dargestellt. Für die Anstastabweichung wurde ein Erwartungswert von $\mu_{PF} = 1,62 \times 10^{-8}$ mm bei einer Standardabweichung von $\sigma_{PF} = 0,39$ mm ermittelt.

Zusätzlich wurde die Durchmesserabweichung PS bestimmt. Sie Ergibt sich aus der Differenz zwischen dem tatsächlichen Durchmesser der Kalibrierkugel D_r und dem berechneten Durchmesser der Ausgleichskugel D_a. Über alle Kugelpositionen im Messvolumen lag die Durchmesserabweichung zwischen $PS_{min} = -1,64$ mm und

$PS_{max} = 1,30$ mm bei einem Erwartungswert von $\mu_{PS} = -0,49$ mm und einer Standardabweichung von $\sigma_{PF} = 0,68$ mm.

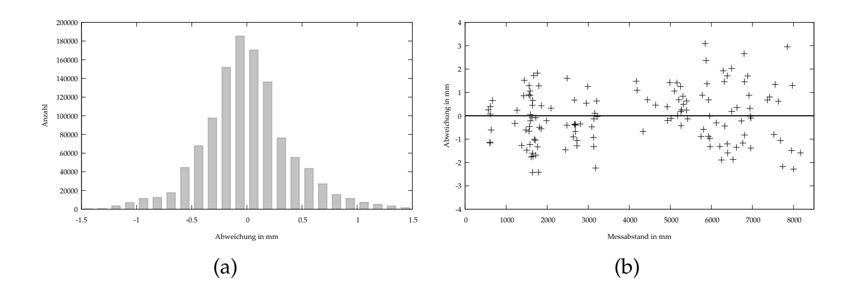


Abbildung 3.1: Ergebnisse der Messungen nach VDI/VDE 2617: (a) Histogramm der Antastabweichung im Intervall von 0,5 mm, (b) Darstellung der Längenmessabweichungen im Verhältnis zu gemessenen Längen.

Die zweite Kenngröße nach VDI/VDE 2617 ist die *Längenmessabweichung*, die das 3D-Abweichungsverhalten im gesamten Messvolumen angibt. Zur Bestimmung wurden wiederum Kalibrierkugeln mit einem Radius von 50 mm so im gesamten Messvolumen des Inspektionssystems verteilt, dass Abstände entlang der Raumachsen und der Volumendiagonalen gemessen werden konnten. Die Referenzwerte wurden ermittelt, indem die Position der Kugeln mit einem Laser-Tracker *Leica AT901 LR* angetastet und die Abstände berechnet wurden. Dieses Verfahren ersetzt den geforderten Längenmaßstab. Die Längenmessabweichung E ergibt sich aus der maximalen Differenz zwischen tatsächlichen und gemessenen Kugelzentren. Zur Ermittlung der Kenngröße wurde die Kalibrierkugel in allen Raumecken und an Zwischenpositionen entlang der Raumkanten positioniert. Insgesamt ergaben sich 12 Positionen, bei denen die Kugel wiederum jeweils aus drei Richtungen mit dem 3D-Sensor erfasst wurde. Eine Ausgleichskugel mit festem Radius wurde in die Messdaten eingepasst und es wurden alle Abstandskombinationen berechnet. Die resultierenden Längenmessabweichungen sind in Abb. 10.1(b) visualisiert.

Die maximale Längenabweichung beträgt im gesamten Volumen $E_{max} = 3,10$ mm bei einem Erwartungswert von $\mu_E = -0,03$ mm und einer Standardabweichung von $\sigma_{PF} = 1,17$ mm.

Bei der Auswertung der Messungen fällt auf, dass das System lokal betrachtet einen negativen Skalierungsfehler besitzt. Der Kugeldurchmesser wird unabhängig von der Position im Messvolumen tendenziell zu klein gemessen. Global betrachtet lässt sich für das gesamte Messsystem feststellen, dass der Längenmessfehler annähernd gleichverteilt ist und nur einen sehr geringer Linearitätsfehler auszumachen ist. Eine mögliche Erklärung dafür ist, dass der Messfehler insbesondere durch das Outside-In-Tracking verursacht wird, welches in dem Volumen unterhalb des Portals statt findet. Die absolute Portalposition, die per Inside-Out-Tracking berechnet wird, auf den Schienen scheint einen eher geringen Einfluss auf die Unsicherheit des Systems zu haben.

4 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein System vorgestellt, mit dem große Bauteile geometrisch auf Abweichungen inspiziert werden können. Die Inspektion erfolgt interaktiv durch Nutzung von Augmented Reality und dimensioneller Oberflächenmessung mit einem fusionierten Kamera-Phasenshift-Sensor. Per Augmented Reality lassen sich recht zügig geometrische Abweichungen erkennen, die dann metrisch mit einem 3D-Sensor gemessen werden können. Die Lage des Sensors wird innerhalb eines Portals durch ein externes, optisches Tracking-System erfasst. Gleichzeitig erfasst das Tracking-System seine eigene Lage. Outside-In und Inside-Out-Tracking werden somit mit einem einzigen System erreicht. Die Systemgenauigkeit ist für das Szenario der Rohteilüberprüfung in der CNC-Bearbeitung ausreichend. Weiter Arbeiten werden sich mit der Optimierung der LED-Anordnung am Sensor beschäftigen, um zu gewährleisten, dass auch unter ungünstigen Verdeckungssituationen stabile Ergebnisse erreicht werden können.

5 Danksagung

Die hier vorgestellten Ergebnisse beruhen auf Arbeiten, die im Forschungsprojekt *Zwanzig20 - Allianz 3Dsensation: 3D-KOSYMA*, FKZ 03ZZ0446H durch das BMBF gefördert wurden. Die Autoren möchten sich für die Unterstützung bedanken.

Literatur

1. W. Chang, J. Hsu, and B. Hsu, "3d scanning system of structured light for aiding workpiece position of cnc machine tool," in *2018 IEEE International Conference on Advanced Manufacturing (ICAM)*, 2018, pp. 388–391.
2. H. Ben Abdallah, I. Jovančević, J.-J. Orteu, and L. Brèthes, "Automatic inspection of aeronautical mechanical assemblies by matching the 3d cad model and real 2d images," *Journal of Imaging*, vol. 5, no. 10, p. 81, 2019.
3. T. Kersten, D. Starosta, and M. Lindstaedt, "Zum genauigkeitspotential aktueller handgeführter 3d-scanner," in *Photogrammetrie, Laserscanning, Optische 3D-Messtechnik - Beiträge der Oldenburger 3D-Tage*, 2018.
4. E. Lilienblum and B. Michaelis, "Optical 3d surface reconstruction by a multi-period phase shift method." *JCP*, vol. 2, no. 2, pp. 73–83, 2007.
5. A. M. Aurand, J. S. Dufour, and W. S. Marras, "Accuracy map of an optical motion capture system with 42 or 21 cameras in a large measurement volume," *Journal of Biomechanics*, vol. 58, no. 1, 2017.
6. P. Eichelberger, M. Ferraro, U. Minder, T. Denton, A. Blasimann, F. Krause, and H. Baur, "Analysis of accuracy in optical motion capture–a protocol for laboratory setup evaluation," *Journal of biomechanics*, vol. 49, no. 10, pp. 2085–2088, 2016.
7. T. Dunker and S. Sperling, "A calibration strategy for systems with 2-d laser sensors," in *10th IMEKO Symposium Laser Metrology for Precision Measurement and Inspection in Industry*, 2011, pp. 265–272.
8. S. Sauer, T. Dunker, and M. Heizmann, "Ein Framework zur Simulation optischer Sensoren," in *20. GMA/ITG-Fachtagung Sensoren und Messsysteme 2019*. Nürnberg: AMA , 2019.
9. VDI Verband Elektrotechnik, Elektronik, Informationstechnik, "VDI-Richtlinie 2617 Blatt 6.2: Genauigkeit von Koordinatenmessgeräten – Kenngrößen und deren Prüfung," 2005.

Extrinsische Kamera zu Lidar Kalibrierung in Virtual Reality

Elias Birkefeld†, Florian Wirth† und Christoph Stiller

KIT, Institut für Mess- und Regelungstechnik (MRT),
Engler-Bunte-Ring 21, 76131 Karlsruhe

Zusammenfassung In dieser Arbeit wurde eine Anwendung für die Bestimmung der extrinsischen Kalibrierungsparameter zwischen einer Kamera und einem Lidar entwickelt. Dies wird anhand einer Projektion des Kamerabildes auf eine Oberflächengeometrie, die aus der Lidar-Punktwolke generiert wurde, umgesetzt. Die Pose der Sensoren relativ zueinander wird manuell mittels VR-Technologie gesetzt. Anschließend werden die sechs Parameter der Kamerapose manuell approximiert, indem Featurepaare in Bild und Punktwolke gefunden und überlagert, gefundene Featurepaare zueinander fixiert und somit gezielt Freiheitsgrade der Kamerabewegung eliminiert werden. Eine erfahrene Testperson erreicht eine Standardabweichung von rund $\pm 0.11m$ und $\pm 0.34°$ innerhalb weniger Minuten. Dementsprechend kann dieses Tool verwendet werden, wenn robuste Initialparameter für die maschinelle extrinsische Kalibrierung benötigt werden und nur eine statische Messung verfügbar ist.

Keywords Kalibrierung, Virtuelle Realität, Tooling, Kamera zu Lidar

1 Einleitung

Mit der Erschließung der Unterhaltungsbranche über das letzte Jahrzehnt haben verbrauchernahe Virtual Reality (VR) Systeme an Popularität zugenommen. Nicht nur Videospiele profitieren von der

† Gleichwertiger Beitrag.

verlustfreien Darstellung dreidimensionaler Szenen, auch bei technischen Anwendungen eröffnet VR neue Möglichkeiten Probleme einfacher, schneller und effizienter zu lösen. Insbesondere bei Problemen, die dreidimensionale Messungen mit hoher Informationsdichte umfassen wie beispielsweise die Kalibrierung einer Kamera zur dreidimensionalen Punktwolke eines Lidar[1]-Sensors, kann VR als Eingabeschnittstelle bisher ungenutzte Potentiale födern. Die Ausmessung der Relativanordnungen von Sensoren, welche als extrinsische Kalibrierung bezeichnet wird, ist von zentraler Bedeutung für die Sensordatenfusion. Die Vorteile der hier vorgestellten manuellen Methode gegenüber herkömmlicher Methoden werden im folgenden Szenario erörtert:

Im Jahre 2040 fahre ein lenkradloses vollautomatisches Taxi auf einer schlechtbeleuchteten Landstraße. Es kommt zu einem Wildunfall, bei der die vordere Stoßstange signifikant deformiert wird. Der an der Stoßstange montierte Lidar verdreht sich in allen Dimensionen. Ohne Neukalibrierung kann die Fahrt nicht fortgesetzt werden, da sonst eine sichere computergestützte Wahrnehmung nicht gewährleistet ist. Die Sensorik muss nun vorort hinreichend genau kalibriert werden.



Abbildung 1.1: In VR visualisierte dreidimensionale Darstellung.

Die im System eingebetteten automatischen Methoden können aus den Sensordaten nicht hinreichend viele visuelle Merkmale extra-

[1] **L**ight **d**etection **a**nd **r**anging, *kurz:* LiDAR oder Lidar, *deutsch:* Lichtdetektion und Abstandsmessung

hieren, um die erforderliche Kalibriergenauigkeit zu erreichen. Die Sensordaten werden daher an ein Servicecenter des Fahrzeugherstellers geschickt, wo ein Servicemitarbeiter mithilfe des hier vorgestellten Tools schnell und routiniert eine initiale Kalibrierung erstellt und zurücksendet. Diese wird während der Weiterfahrt algorithmisch optimiert.

Die in dieser Arbeit vorgestellte Anwendung soll einen Einblick in die Anwendungsmöglichkeiten von Virtual Reality im Bereich Kalibrierung zeigen. In der Regel werden digitale Repräsentationen dreidimensionaler Objekte auf zweidimensionale Bildschirmflächen projiziert, woraus eine Mehrdeutigkeit resultiert. Ein modernes Virtual Reality System ermöglicht eine intuitive Darstellung dreidimensionaler Szenen, indem es diese Mehrdeutigkeit durch stereoskopische Darstellung umgeht. Diese Vorteile wurden bereits durch das Label Tool PointAtMe [1] gezeigt, das der manuellen Generierung von Objektannotationen in Punktwolken dient.

2 Stand der Wissenschaft

Kalibrierung von Sensoren ist ein altes Forschungsgebiet, das mit der Entwicklung neuer Sensoren und Messprinzipien aktuell bleibt. Bei der Kalibrierung eines Sensors werden zwei Teilbereiche unterschieden: Das intrinsische Sensormodel projiziert den realen Sensor auf ein mathematisches Modell; die extrinsischen Kalibrierung besteht aus Translation und Rotation, die der Sensor relativ zu einem Bezugskoordinatensystem aufweist.

Extrinsische Parameter können basierend auf Kalibrierobjekten mit bekannter Geometrie oder ohne Kalibrierobjekte bestimmt werden. Für die objektbasierte Kalibrierung werden Objekte wie bspw. Schachbrettmuster [2], Kugeln [3] oder Kreise [4] im Sichtfeld aller zu kalibrierenden Sensoren platziert. Diese werden algorithmisch detektiert. Durch die Position und Orientierung des Objektes im Sichtfeld jedes einzelnen Sensors können Rückschlüsse auf die Ausrichtung der Sensoren zueinander gezogen werden. Dies setzt ein Kalibrierobjekt bekannter Geometrie und in der Regel eine Vielzahl von Aufnahmen voraus. Von Geiger *et al.* [2] werden intrinsische und extrinsische Parameter durch das automatische Detektieren mehre-

rer Schachbrettmuster im überlappenden Lidar- und Kamerasichtfeld bestimmt. Heng *et al.* [5] bestimmen intrinsische Kameraparameter per Schachbrettmuster, extrinsische Parameter werden durch Odometrie und Bewegungserkennung im jeweiligen Kamerasichtfeld bestimmt.

Im Falle einer unkontrollierten Umgebung wurden objektunabhängige Methoden entwickelt. Diese basieren beispielsweise auf der Extrahierung von primitiven optischen Merkmalen wie Kanten aus Aufnahmen der zu kalibrierenden Sensoren, welche durch Minimierung einer Kostenfunktion wie von Moghadam *et al.* [6] oder Kang *et al.* [7] vorgeschlagen überlagert werden. Diese Methoden funktionieren in strukturreicher Umgebung mit Beleuchtung, Gebäuden, Pfeilern, Pfosten und Straßenmarkierungen präzise und robust. Für kantenarme Regionen, wie sie auf Überlandstraßen durch Wälder und über Felder vorzufinden sind, büßen sie an Genauigkeit ein. Ansätze wie von Taylor *et al.* [8] sind auf ein problemspezifisches Umfeld angepasst und somit nicht universell einsetzbar. Manuelle Methoden wie von Scaramuzza *et al.* [9] funktionieren dagegen für beliebige Szenen zuverlässig.

Für die Projektion eines Bildes auf ein dreidimensionales Objekt wird eine Projektionsflache benötigt. Es gibt bereits Verfahren, die sich mit der Oberflächengenerierung aus Punktwolken beschäftigen [10, 11]. Allerdings konzentrieren sich diese Verfahren auf die Generierung einer lückenfreien Oberfläche. In dieser Arbeit sollen Bereiche der Punktwolke mit wenigen Punkten aber explizit als Loch dargestellt werden, damit sich der Nutzer nicht an automatisch generierten Oberflächen orientiert, sondern nur an tatsächlich eingemessenen.

3 Ansatz

3.1 Oberflächengenerierung aus Punktwolken

Die Oberfläche soll aus so wenigen Polygonen wie möglich bestehen, um Echtzeitfähigkeit zu gewährleisten.

In einem ersten Schritt werden Punkte zusammengefasst, die nahe beieinander liegen. Hierfür werden räumliche Sektoren s_{ϕ_i,θ_j} für

$i \in [1,..,n_s]$ und $j \in [1,..,n_z]$ eingeführt, die Winkelabschnitten in Kugelkoordinaten ϕ, θ, r im Lidarsichtfeld entsprechen. n_s ist die gewählte horizontale Auflösung, n_z ist die Zeilen-/Diodenanzahl des Lidars. Alle Punkte, die nach Projektion auf diese Kugel im selben Sektoren liegen, werden als Punktgruppe zusammengefasst. Ist die größte Distanz eines Punktes in dieser Punktgruppe zu einem anderen kleiner als ein Schwellwert $d_{\text{Limit, PG}}$, werden die Koordinaten all dieser Punkte gemittelt und dem entsprechenden Sektoren zugeordnet.

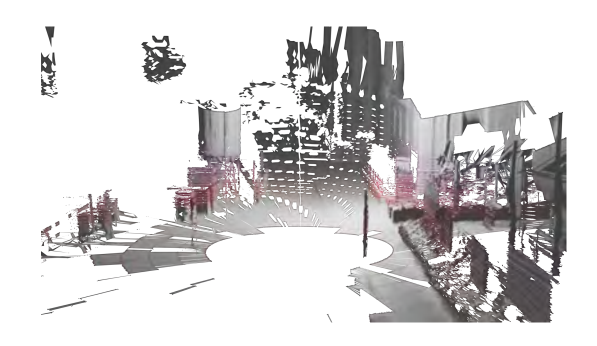


Abbildung 3.1: Generierte Oberfläche aus einer Punktwolke.

Ist die Distanz größer, wird die Punktgruppe unterteilt und die Schwellwertbedingung erneut überprüft. Für jede so unterteilte Punktgruppe werden die einzelnen Koordinaten der Untergruppen nicht gemittelt sondern die äußersten Punkte extrahiert, um Kanten in der Punktwolke zu erhalten. Idealerweise wird die Größe und die damit verbundene Anzahl der Sektoren so gewählt, dass pro Sektor maximal zwei Punktgruppen vorliegen.

Im zweiten Schritt sind Polygone aus den verbleibenden Punkten zu generieren. Es werden jeweils vier Sektoren s_{ϕ_i,θ_j}, $s_{\phi_i,\theta_{j+1}}$, s_{ϕ_{i+1},θ_j}, $s_{\phi_{i+1},\theta_{j+1}}$ überprüft. Im einfachsten Fall enthält jeder Sektor genau eine Punktgruppe. Dann werden zwei Dreiecke gebildet, wenn für den

Normalenvektor $n_{i,j}$ des jeweiligen Dreiecks und die Verbindungsgerade $l_{i,j}$ vom Lidar zur Punktgruppe des Sektors s_{ϕ_i,θ_j}

$$\arccos\left(\frac{s_{\phi_i,\theta_j} \cdot l_{i,j}}{|s_{\phi_i,\theta_j}| \cdot |l_{i,j}|}\right) \geq 90^\circ - \alpha_{\text{Limit}} \tag{3.1}$$

gilt, wobei $\alpha_{\text{Limit}} < 5^\circ$ gewählt wird. Falls in mindestens einem Sektor mehr als eine Punktgruppe vorhanden ist, wird jede mögliche Punktekombination in Betracht gezogen, die aus drei Punkten in unterschiedlichen Sektoren besteht. Jede aus diesen Kombinationen entstehenden Dreiecke, die die obige Bedingung erfüllen, werden generiert. Ein Beispiel für eine so generierte Oberfläche und die zugehörige Punktwolke ist in Abb. 3.1 zu sehen.

3.2 Projektion des Bildes auf die Oberfläche

Das Bild wird mithilfe der Projektionsmatrix des Lochkameramodells auf die generierte Oberfläche projiziert. Bisher wird das Lochkameramodell unterstützt. Die Projektion wird mithilfe eines Shaders durchgeführt, der parallelisiert und somit auch für große Datenmengen in Echtzeit auf einer Grafikkarte berechnet wird. Die Oberflächen ermöglichen eine gegenseitige Abschattung, wodurch verhindert wird, dass der gleiche Bildausschnitt auf mehreren Oberflächen abgebildet wird.



Abbildung 3.2: Auf die Oberfläche projiziertes Kamerabild.

3.3 Ankerpunkte

Das wichtigste der zahlreichen Hilfsmittel, die dem Nutzer zur Verfügung gestellt werden, sind Ankerpunkte, die Freiheitsgrade der Kamerapose beschränken und so den Kalibriervorgang beschleunigen. Es können bis zu zwei Ankerpunkte gesetzt werden. Diese Anker unterbinden eine Verschiebung des projizierten Bildes an dem Punkt, an dem sie in die Punktwolke gesetzt werden.

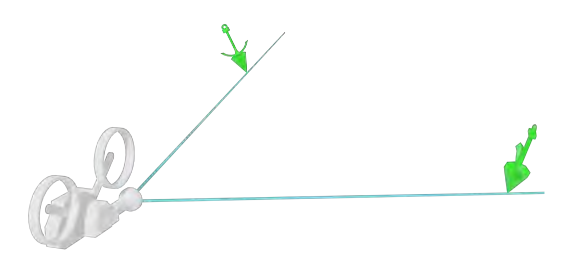


Abbildung 3.3: Wenn die grünen Anker gesetzt werden, verlaufen zwei Sichtstrahle der Kamera gezwungenermaßen durch die Anker. Dies schränkt die Freiheitsgrade der Kamerapose ein.

Wenn ein Anker gesetzt ist, werden zwei (Rotations-) Freiheitsgrade eliminiert. Wenn zwei Ankerpunkte gesetzt sind, können nur noch zwei (Translations-) Freiheitsgrade verändert werden. In beiden Fällen ist keiner der Parameter der Kamerapose konstant, wird aber in Abhängigkeit der verbleibenden Freiheitsgrade angepasst. Abb. 3.3 zeigt beispielhaft zwei Anker und die zugehörigen Sichtstrahle.

3.4 Eingabeschnittstelle

Es wird das VR Gerät Oculus Rift verwendet, das Touch Controller mit je 6 Freiheitsgraden und mehrern Tasten bereitstellt. Anhand der Controller können sowohl die virtuellen Sensoren, als auch die Anker gegriffen und neu positioniert werden.

4 Evaluation

4.1 Sensorsetup

Das Tool wurde anhand des Sensorsetups des teilautomatischen Versuchsfahrzeugs des MRT evaluiert. Die verwendete Kamera ist vom Typ FLIR BlackFly S 9 MPx mit einem Weitwinkelobjektiv mit rund 120° Öffnungswinkel. Das Rohbild wurde auf ein Lochkameramodell umgerechnet. Der verwendete Lidar ist vom Typ Velodyne VLS-128 Alpha Prime.

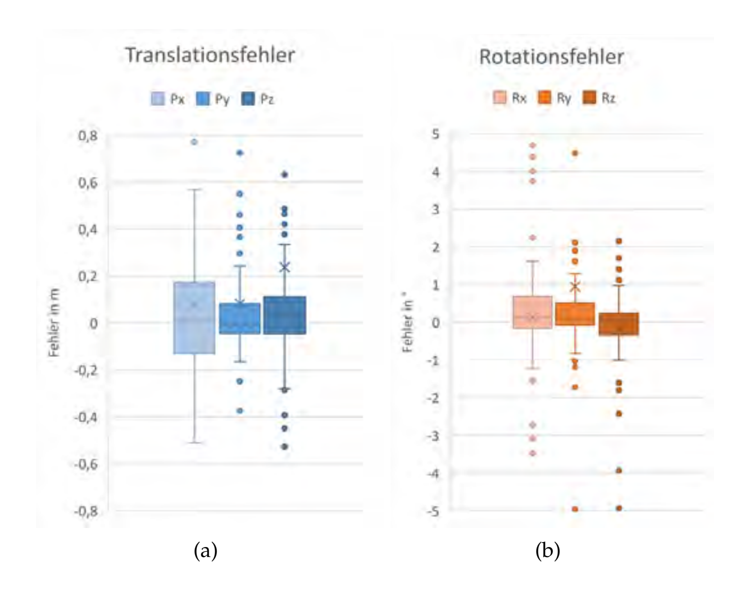
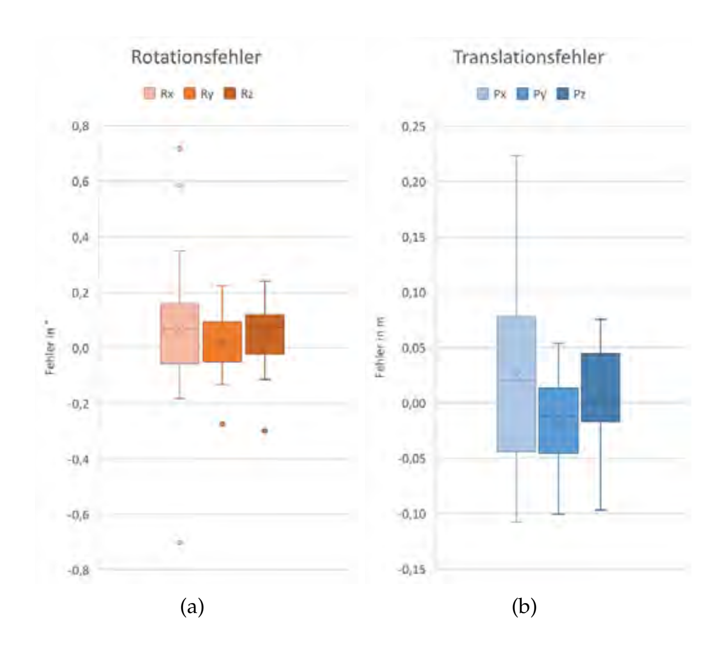


Abbildung 4.1: Ergebnisse aller Probanden über alle Szenarien.

4.2 Probandenexperimente und Randbedingungen

Die Testszenen stammen von 21 diversitären Verkehrszenarien. Vier Probanden wurden gebeten auf allen 21 Szenarien die extrinsische Kalibrierung mithilfe des vorgestellten Tools zu erstellen. Die Probanden bekamen keine Rückmeldung bezüglich ihrer aktuellen Kalibriergenauigkeit. Für alle Versuche wurde die gleiche maschinen-



(a) (b)

Abbildung 4.2: Ergebnisse des routiniertesten Probanen über alle Szenen.

erstellte Kalibrierung nach Strauss *et al.* [12] als Referenz verwendet. Aufgrund eines Rotationsfehlers um die Nick-Achse wurde die Rotation in dieser Komponente angepasst. Ob dieser Rotationsfehler aus der Umrechnung der Koordinatensysteme herrührt oder die wahre Kalibrierung tatsächlich eine Ungenauigkeit besitzt, konnte nicht geklärt werden. Nach dieser minimalen Anpassung kann man sich in VR davon überzeugen, dass Bild und Punktwolke augenscheinlich besser zueinander passen.

4.3 Ergebnisse

In Abb. 4.1 ist der Translationsfehlervektor aufgeteilt in x, y und z Anteile, sowie die Eulerwinkel um die x, y und z Achse aufgetragen. Hierbei entspricht die x-Achse der optische Achse, die y-Achse zeigt in Fahrtrichtung rechts und die z-Achse entsprechend nach oben.

Erwartungsgemäß findet sich der größte Fehler in Richtung der optischen Achse.

Als besondern klein stellen sich die Rotationsfehler heraus, die Translationsfehler sind vermutlich sogar größer als mit einfachsten Messinstrumenten erzielbar, womit Fehler unterhalb von 10 cm problemlos erreichbar sein sollten. Dennoch stellt das vorgestellte Tool für den eingangs geschilderten Anwendungsfall eine schnelle Fernkalibrierung als Servicemaßnahme eine geeignete, innovative Lösung dar, insbesondere aufgrund der geringen Rotationsfehler.

Des Weiteren sind die Ergebnisse in Abb. 4.2 zu beachten, die vom geübtesten der Probanden unter Zuhilfenahme der Anker erzeugt wurden. Hierbei lässt sich erkennen, dass durch den erfahrenen Umgang mit dem Tool deutliche Verbesserungen im Vergleich zu den Laienprobanden erzielt werden können.

5 Zusammenfassung

In der Arbeit wird ein manuelles Kalibrierwerkzeug vorgestellt, bei dem die Möglichkeiten von Virtual Reality als Hilfsmittel zur Lösung technischer Aufgabenstellungen am Anwendungsfall der extrinsischen Sensorkalibrierung erprobt werden soll. Aus einer Lidar-Punktwolke wird eine 3D Oberfläche in VR generiert, auf welches anhand des Lochkameramodells ein Kamerabild projiziert wird. Umfassende Tools zur leichteren Handhabung wurden integriert, mit welchen die Relativposition der Sensoren zueinander dem angepasst werden kann. Anhand einer Probandenstudie wurde die erzielbare Genauigkeit festgestellt. Demnach kann das Tool vor allem den Rotationsfehler stark verringern. Um eine schnelle, datengestützte Fernwartung zu bewerkstelligen, liefert das Tool gute Initialwerte, die danach bei Weiterfahrt algorithmisch verbessert werden können.

Literatur

1. F. Wirth, J. Quehl, J. Ota, and C. Stiller, “PointAtMe: Efficient 3D Point Cloud Labeling in Virtual Reality,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1693–1698.

2. A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3936–3943.
3. T. Kühner and J. Kümmerle, "Extrinsic Multi Sensor Calibration under Uncertainties," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3921–3927.
4. H. Alismail, L. D. Baker, and B. Browning, "Automatic calibration of a range sensor and camera system," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, 2012, pp. 286–292.
5. L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1793–1800.
6. P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3685–3691.
7. J. Kang and N. L. Doh, "Automatic targetless camera–LIDAR calibration by aligning edge with Gaussian mixture model," *Journal of Field Robotics*, vol. 37, no. 1, pp. 158–179, 2020.
8. Z. Taylor and J. Nieto, "A mutual information approach to automatic calibration of camera and lidar in natural environments," in *Australian Conference on Robotics and Automation*, 2012, pp. 3–5.
9. D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 4164–4169.
10. N. Amenta, M. Bern, and M. Kamvysselis, "A new Voronoi-based surface reconstruction algorithm," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 415–421.
11. M. Kolahdouzan and C. Shahabi, "Voronoi-based k nearest neighbor search for spatial network databases," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, 2004, pp. 840–851.
12. T. Strauß, J. Ziegler, and J. Beck, "Calibrating multiple cameras with non-overlapping views using coded checkerboard targets," in *17th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 2014, pp. 2623–2628.

Minimal Paths for 3D Crack Detection in Concrete

Franziska Müsebeck[1], Ali Moghiseh[2], Claudia Redenbach[1] and Katja Schladitz[2]

[1] Department of Mathematics, University of Kaiserslautern
Gottlieb-Daimler-Straße 47, 67663 Kaiserslautern
[2] Fraunhofer-Institut für Techno- und Wirtschaftsmathematik,
Fraunhofer-Platz 1, 67663 Kaiserslautern

Abstract Concrete is a commonly used construction material for buildings, bridges and roads. As safety is very important in such constructions, the investigation of damage processes in concrete is a highly relevant topic. For instance, the early detection of cracks can prevent the collapse of a bridge. Thus, the necessity of automated crack detection and segmentation arises. We generalize and modify a 2D method for crack detection in road pavement images to 3D. It is based on modeling the image as a graph and searching for minimal paths therein. The proposed 3D method is evaluated on synthetic crack images and applied to a 3D computed tomography image of real concrete.

Keywords Crack detection, image segmentation, three-dimensional, computed tomography

1 Introduction

The investigation of damage processes in concrete has become an increasingly popular topic. For design, monitoring and maintenance of buildings or other constructions, it is essential to understand damage of building materials. In order to gain a deeper understanding of the mechanical properties of concrete, crack initiation and development are studied on concrete specimens during loading tests using

computed tomography (CT). Visual inspection of such a 3D CT image and manual segmentation of cracks is a time intensive process and can therefore only be executed on a few slices as these images are usually very large [1]. This makes automated detection of cracks in 3D images desirable. While crack segmentation in 2D images has already been widely researched, so far, there is no satisfying solution available in 3D [1].

A typical motivation to look for cracks in 2D images is monitoring of the road surface conditions by identifying cracks in pavement images of roadways. There is a variety of crack detection methods based on two general assumptions. The first one concerns the brightness, i. e., the crack is darker than the background which means that the gray values of the crack pixels are smaller compared to the neighboring background intensities. The second one is related to geometry, namely that the crack is continuous which means that the crack pixels are connected.

Here, we present a generalization of the 2D approach in [2] which uses exactly these characteristics. The image is modeled as a graph and minimal paths are computed. If the nodes are weighted by the pixels' gray values, a minimal path possesses the above mentioned crack properties: it is connected and consists of vertices with small weights, i. e. small intensity values.

We first introduce the 2D method proposed in [2] which we will refer to as *MinPath2D* and then present our approach of a generalization to 3D images. Adjustments are made on the one hand to obtain an appropriate image model and on the other hand to save computation time.

2 2D cracks by minimal paths

Image model

The MinPath2D algorithm uses a representation of the gray value image as a directed, vertex-weighted graph. In fact, several graphs can be defined for one image depending on the directions represented in the 8-neighbourhood on the pixel grid. In all these graphs, the set of vertices corresponds to the pixels in the image and the weights are the pixels' intensities. The graphs differ by the set of

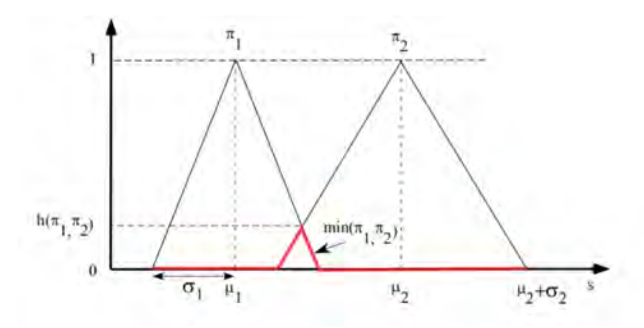


Figure 2.1: Sup-min function to evaluate the degree of coherence of two information sources [2]

edges which is chosen depending on direction. For each direction, e. g. *up*, a set of three discrete directions is selected. Edges are constructed by connecting all vertices to their three neighbors towards these. For instance, in the graph for the direction *up*, a vertex is connected to its neighbor directly above as well as to the two vertices left and right from the vertex above. This way, we can construct directed, vertex-weighted graphs for the eight directions in 2D: *up*, *down*, *left*, *right*, *upper left*, *lower right*, *lower left*, *upper right*.

Searching minimal paths

Based on this image model, the MinPath2D algorithm computes minimal paths in the above described graphs. The algorithm gets a graph $G = (V, E)$ and a predefined path length ℓ as inputs and computes for each vertex $x \in V$ a path with minimal weight of length ℓ starting in the considered vertex x. Applying this algorithm to all graphs yields eight paths for each vertex. The two paths for opposite directions are merged into one path of length $2\ell + 1$, whereby the start pixel of the individual paths becomes the center pixel of the merged path. This procedure results in four minimal paths for each pixel.

Pixelwise classification

In the next step, it has to be decided whether or not a pixel belongs to a crack. The basic idea is to introduce a characteristic which takes a small value in an orientation along the crack and higher values along other orientations. We use the Free Form Anisotropy (FFA) measure which was derived from possibility theory [3]. This theory based on fuzzy sets is used to model uncertainty by introducing a degree of membership for the elements of a set. Possibility distributions are modelled by fuzzy set membership functions. For possibility theory, they play the role of probability densities for probability theory. The relationship between probability theory and possibility theory has been discussed as both theories seem to be similar in the sense that both deal with some type of uncertainty and both use the $[0, 1]$ interval as range of their respective functions [4]. It is however difficult to compare them as it is not clear on which level – e. g. mathematically, semantically, or linguistically. See [4] for details.

Here, we use the conversion of each minimal path found in the first step into a so-called information source π_i [5] which is represented by a possibility distribution. To this end, the mean value and the standard deviation of gray levels of each path are computed. Figure 2.1 shows two partially overlapping possibility distributions. This means, that certain gray values are considered possible by both information sources. The degree h of coherence between the two information sources is measured by the sup-min function. The greater the intersection, the higher the degree of coherence. This concept is applied to the classification problem.

If a pixel belongs to a crack, then among the minimal paths found in the first step, at least the path with smallest mean intensity follows the crack. Comparing the information source of this path with another one corresponding to a path running in the background, we observe a small coherence h. This is due to the fact that the number of gray levels occurring in both paths is rather small. In contrast, the coherence h of two different background sources is comparatively large since the gray level distributions of the two corresponding paths are more similar. The Free Form Anisotropy of a pixel x is then defined as the degree of conflict between information sources $\pi_{\min} = (\mu_{\min}, \sigma_{\min})$ and $\pi_{\max} = (\mu_{\max}, \sigma_{\max})$ of the two paths with

center x and minimal and maximal mean gray value, respectively, i. e.

$$\mathrm{FFA}(x) = 1 - h(\pi_{\min}, \pi_{\max}) = 1 - \sup \min(\pi_{\min}, \pi_{\max}).$$

The FFA measure takes values between 0 and 1 and is close to 1 if x is a crack pixel. Thus, the final algorithm has two parameters, the length parameter ℓ and a threshold $t \in [0, 1]$ for which all pixels x with $\mathrm{FFA}(x) < t$ are classified as background and pixels with $\mathrm{FFA}(x) \geq t$ are identified as crack pixels.

In practice, we calculate the coherence h according to Figure 2.1 as intersection of two lines L_1 and L_2, where L_1 is passing trough the two points $(\mu_1, 1)$ and $(\mu_1 + \sigma_1, 0)$ and L_2 is passing trough $(\mu_2, 1)$ and $(\mu_2 - \sigma_2, 0)$.

3 3D cracks by minimal paths

Image model

The image model is constructed in a similar way as in 2D. In particular, we use a 3D graph, where the set of vertices corresponds to the voxels in the image which are weighted by their gray values. The definition of the set of edges leaves more room for discussion. The number of direct neighbors of one voxel increases with dimension. An inner voxel has 26 neighbors in 3D while in 2D there are only eight neighbors. As a consequence, there are 26 directions implying 13 different orientations. The question arises, how many and which orientations we want to consider in the algorithm and how to define the discretization of one direction.

Directions can be categorized into three groups, where the first consists of the three main orientations: *up* and *down*, *left* and *right*, *in front* and *behind*. Secondly, there are the diagonals of a coordinate plane: *upper left* and *lower right*, *upper right* and *lower left*, *in front left* and *behind right*, *in front right* and *behind left*, *in front up* and *behind down*, *in front down* and *behind up*. The last group is formed by space diagonals (connecting two vertices that are not in the same coordinate plane): *upper left behind* and *lower right in front*, *upper right behind* and *lower left in front*, *upper left in front* and *lower right behind*, *upper*

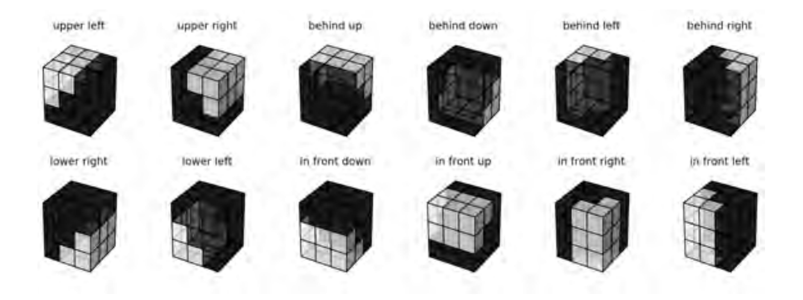


Figure 3.1: Diagonal directions in 3D illustrated for the center voxel

right in front and *lower left behind*. We will only consider directions from the first and second category. This restriction is justified by saving computational time and moreover makes sense since the FFA measure only uses two paths and thus only two orientations to perform the classification. The decision to neglect the third category is based on the fact that this is the category which has the most overlap of voxels with the other categories.

The natural extension of the discretization is to take nine discrete directions into account for each direction. For instance, in 2D there are three pixels above an inner pixel in the same way as there are nine voxels above an inner voxel in 3D. Thus, we define the set of discrete directions for one direction by nine discrete directions as illustrated for the diagonals in Fig 3.1. Summarizing, a neighborhood for one voxel and for one direction is determined by the nine neighbors towards that direction. Moreover, we choose to take nine different orientations into account, namely the three main orientations and the six plane diagonals.

Searching local minimal paths

Due to simplicity and the increasing computational effort in 3D, we deviate from the 2D method when calculating minimal paths by using a greedy propagation algorithm that returns only a locally minimal path. Beginning from the start node, the neighbor with the smallest weight is successively added to the path. This local propa-

gation does not necessarily yield a path with total minimal weight, but in the case of a crack voxel as start voxel, the resulting path still follows the crack path approximately and thus the classification by assessing the degree of coherence still works. Ultimately, it is more important that the different orientations are represented than that the path actually has minimal weight.

Voxelwise classification

The classification per voxel is performed as in the MinPath2D algorithm. However, we use the coherence h (rather than the FFA measure) to compare two information sources. Then, a threshold t is chosen in $[0, 1]$ such that all values less than or equal to t are classified as crack voxels and all values above are classified as background.

Post-processing

The output of the classification algorithm shows some discontinuities where single voxels are missing in the crack surface, see Section 4, Figure 4.1. Hence, we apply a morphological closing followed by an erosion [6] to the binary classification output to connect the crack pixels and to remove single falsely identified background pixels.

Parameter selection

As described above, our algorithm has two parameters, the length parameter ℓ and the threshold t which must be selected appropriately. The influence of the parameters was investigated experimentally by keeping one fixed while varying the other one. We observed that the length parameter ℓ should be chosen in the interval $[24, 48]$ depending on the resolution of the image. The choice of the second parameter, the threshold t, turns out to be less important than the choice of ℓ as long as it is sufficiently small. A good choice seems to be $t \in [0.001, 0.1]$.

4 Application

Evaluation metrics

For the evaluation on the synthetic crack images, we use the F1-measure as an overall accuracy measure which is calculated from precision and recall.

The Precision is defined as

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{ false positives}}$$

and measures how exact a positive result is, i. e. what proportion of voxels classified as positive are indeed positive.

The Recall is defined as

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

and measures the proportion of positives which were identified correctly.

The F1-measure can be interpreted as a weighted average of Precision and Recall taking both aspects into account. It is given by

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

All three measures take values between 0 and 1 and reach their best value at 1. In practice, it is impossible to exactly define the crack boundary in real images. Hence, it is not uncommon to introduce a certain pixel (voxel) tolerance in the evaluation of a segmentation algorithm. In the following, we evaluate the results allowing for a tolerance of falsely classified voxels within a certain distance of the true crack. In our case, using a tolerance of tol voxels has the following meaning:

- A false negative voxel is counted as true positive, if in the predicted image there is a voxel classified as positive within a distance of tol.
- A false positive voxel is counted as true positive, if in the ground truth there is a voxel classified as positive within a distance of tol.

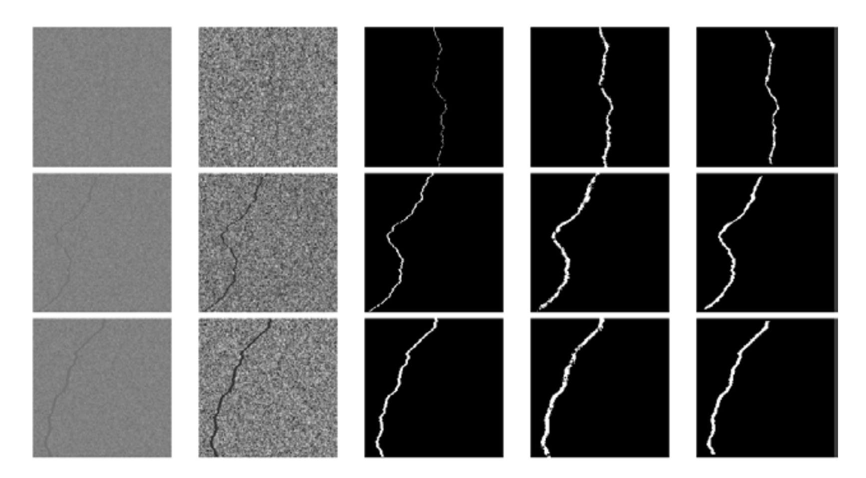


Figure 4.1: Z-slices of the results of the application on 256^3 synthetic 3D images with $l = 24$ and $t = 0.01$. From left to right: input image, normalized input image, ground truth (generated by a realization of the fBS), output before post-processing, output after post-processing

Results on synthetic images

Synthetic cracks are simulated by a realization of a fractional Brownian Surface (fBS) [7] which provides a ground truth by which we can evaluate the output of the algorithm. Given the ground truth as binary crack image (generated by [8]), a corresponding realistic gray value crack image is obtained by generating noisy gray levels for background and crack. Varying crack widths can be generated by dilating the crack. We evaluate the algorithm on synthetic crack images with a crack width of $w = 1$, $w = 3$, and $w = 5$ voxels. Z-slices of the resulting segmentations are shown in Figure 4.1. Figure 4.2 shows the corresponding 3D renderings. The crack is properly segmented in all three cases. However, the detected crack is thicker than the true one. This is especially noticeable when the original crack is very thin as in the case $w = 1$. Evaluating the output by the F1-measure with a tolerance of $tol = 1$, we obtain a value of 0.9609 for the image with $w = 1$, 0.9599 for $w = 3$ and 0.9744 for $w = 5$.

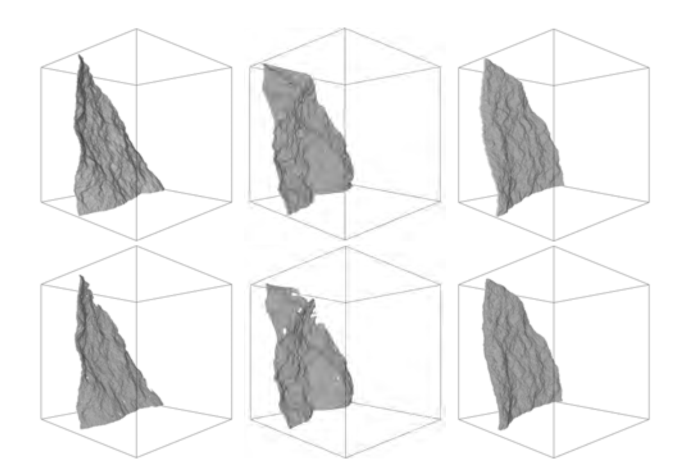


Figure 4.2: 3D renderings of the results on 256^3 synthetic images with $\ell = 24$ and $t = 0.01$. Top row: ground truth. Bottom row: output after post-processing. From left to right: $w = 1$, $w = 3$, $w = 5$.

Results on real images

In the next step, we apply our algorithm to 3D images of real concrete. In particular, we use a 3D CT image of a concrete specimen with a glass fiber reinforced polymer bar used as reinforcement. Pulling out the bar generates cracks. The sample has a diameter of 4.8 cm and the original image has a size of $1\,986 \times 1\,986 \times 1\,576$ voxels. After shrinking the CT image by a factor of two and cutting out a cubic patch, we obtain an image of size 620^3 which we pass as input to the algorithm. Depending on the mixing conditions and the coarseness of the aggregates, the concrete matrix may appear rather heterogeneous in CT images. In our case, for instance, we observe pores as well as highly absorbing grains (see Figure 4.3). The real concrete is clearly more complex than synthetic data. Nevertheless, our algorithm can handle the background structure and is able to segment the crack except for a few really thin and fine structures. Z-slices of the output of our algorithm are shown in Figure 4.3.

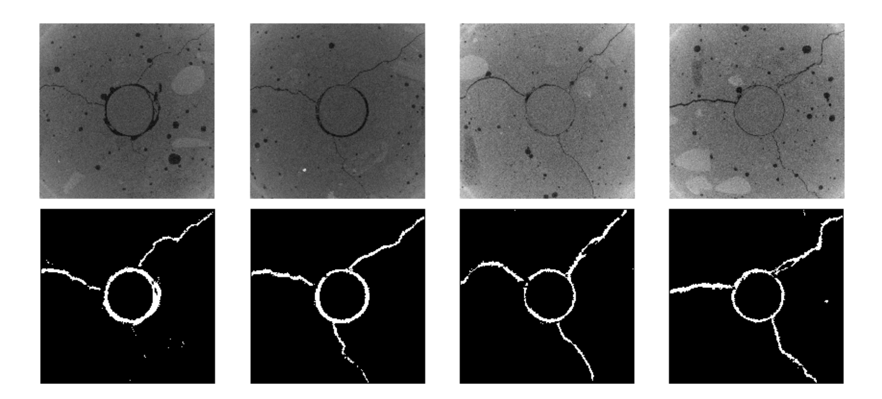


Figure 4.3: Z-slices of the results for a 3D CT image of real concrete using the parameters $\ell = 48$ and $t = 0.001$. Top row: normalized input. Bottom row: output after post-processing. Sample: C. Caspari, University of Kaiserslautern, CT imaging: F. Schreiber, Fraunhofer ITWM

Implementation and runtime

The algorithm was implemented in C++. It can be integrated as a plug-in into the image processing software *ToolIP* (*Fraunhofer-Institut für Techno- und Wirtschaftsmathematik*, [9]) which is used to segment the crack images. The runtime increases with the image size as well as with increasing parameter ℓ. It is roughly 3 minutes for a 256^3 image with $\ell = 24$ and about 30 minutes for a 620^3 image with $\ell = 48$. All runtimes are measured on a standard computer of the image processing department of the ITWM with eight processors of the type Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz distributed on four cores, with a working memory of 16 GB.

5 Summary

We propose a new algorithm for crack detection in 3D images. The method is based on the 2D algorithm from [2] which we generalized to the 3D case. Our generalization includes the adaption of the image modeling as 3D graph, whereby the definition of the neighborhood relation of the vertices by directions as well as the definition and

choice of directions (and orientations) were adapted. Further adjustments were made as we propose the usage of approximate minimal paths obtained by a local propagation algorithm. We demonstrated that the algorithm is able to segment cracks in 3D images. In particular, we applied the algorithm to synthetic images whereby we were able to achieve F1 values between 0.9599 and 0.9744 for test images with different crack width. Moreover, the algorithm is able to deal with images of real concrete and to correctly segment the cracks. Even the more complex background structure containing dark pores can be handled. The main weakness of the algorithm is that the segmented crack is too thick compared to the original one and that it misses some very thin and fine crack structures.

Acknowledgments

This research was supported by the German Federal Ministry of Education and Research under project *Detektion von Anomalien in großen räumlichen Bilddaten* (DAnoBi) and by the Rhineland-Palatinate research initiative through project "Mathematics applied to real world challenges" (MathApp).

We thank Christian Caspari (Department of Civil Engineering, University of Kaiserslautern) for the concrete sample and Franz Schreiber (Fraunhofer ITWM) for the CT imaging.

References

1. O. Paetsch, "Possibilities and limitations of automatic feature extraction shown by the example of crack detection in 3d-ct images of concrete specimen," in *9th Conference on Industrial Computed Tomography (iCT) 2019*, iCT 2019 Conference Proceedings, 2019.
2. M. Avila, S. Begot, F. Duculty, and T. S. Nguyen, "2d image based road pavement crack detection by calculating minimal paths and dynamic programming," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 783–787.
3. L.A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 100, pp. 9 – 34, 1999.

4. J.-H. Zimmermann, *Fuzzy Set Theory—and Its Applications*. Springer, Dordrecht, 2008.
5. I. Bloch, *Information Fusion in Signal and Image Processing: Major Probabilistic and Non-Probabilistic Numerical Approaches*, ser. ISTE. Wiley, 2008.
6. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson Education, 2009.
7. P. S. Addison, L. T. Dougan, A. S. Ndumu, and W. M. Mackenzie, "A fractional Brownian motion model of cracking," in *Paradigms Of Complexity: Fractals And Structures In The Sciences*, 2000, pp. 117–123.
8. Z. Botev, "Fractional Brownian field or surface generator," https://de.mathworks.com/matlabcentral/fileexchange/38945-fractional-brownian-field-or-surface-generator, retrieved June 10, 2020.
9. Fraunhofer ITWM, "Toolip," https://www.itwm.fraunhofer.de/toolip, retrieved June 10, 2020.

Advances in deflectometric form measurement

Marcus Petz, Hanno Dierke, and Rainer Tutsch

Technische Universität Braunschweig, Institut für Produktionsmesstechnik, Schleinitzstraße 20, 38106 Braunschweig

Abstract Phase measuring deflectometry is an accepted technique for measuring the shape of specular surfaces. While defloctometry is known to provide high sensitivity in the nanometer range, the absolute form measuring accuracy is typically inferior by several orders of magnitude. The comparatively low accuracy of typical implementations of phase measuring deflectometry is determined by several influencing factors. On the one hand, many system models used do not consider all relevant system parameters, such as refraction in the display substrate or its flatness deviation. On the other hand, due to the complex system geometry, many calibration procedures are susceptible to deviations due to low condition numbers of the mathematical problems. To increase the absolute accuracy of phase measuring deflectometry, the authors have analyzed in detail the calibration procedures, the measurement process, and the evaluation algorithms and have made numerous extensions and optimizations. The present contribution gives an overview of the obtained findings and the applied measures. The performance of the approach is evaluated based on measurements of challengingly curved measurement objects. Based on these selected objects, form measurement deviations of better than 1 µm are documented.

Keywords Deflectometry, specular surface, phase shifting, structured illumination, system calibration, photogrammetry

1 Introduction

Phase measuring deflectometry is accepted as a highly sensitive method for full-field form measurement of specular surfaces. However, the accuracy of the absolute form measurement has so far significantly lagged behind the sensitivity due to systematic influences. As part of the work presented here, the deflectometric calibration, measurement, and evaluation processes were subjected to a number of extensions and optimizations in order to obtain an absolute form measurement accuracy in the sub-micrometer range for typical optical functional surfaces with diameters of 50 mm to 100 mm.

Deflectometry is based on the observation of reference patterns whose images are reflected by the surface under test and are thereby distorted depending on the surface geometry. In most practical applications of phase measuring deflectometry liquid crystal displays are utilized to represent the required reference patterns. A main concern is therefore to optimize the spatial coding strategies and to characterize the non-ideal display properties, such as characteristic curve, topography, and refractive power.

Another important aspect is the geometric calibration of the whole setup, especially the relative orientation of measuring camera and liquid crystal display. As typically the camera is not able to directly observe the display without the beam deflection of a specular object, the calibration process requires a specular calibration object. Ideally, the properties of that object, especially its spatial location and orientation, are known. However, approaches with a high degree of self-calibration, similar to the established bundle adjustment techniques used in protogrammetry, have also been reported in the literature. The authors have therefore investigated the potential of approaches with a high degree of freedom and have compared their results with information obtained from other strategies.

The surface reconstruction in deflectometry involves the integration of the measured surface gradients. Furthermore, this process requires additional information to regularize the deflectometric problem. In the performed work, the approach of using at least two different relative orientations of object and reference pattern was used, which has been proposed in [1]. But even then, there are several different strategies for processing the obtained measurement data.

In the following, numerous aspects of the deflectometric calibration, measurement, and evaluation processes are discussed as an overview and hints to realizations are given that have proven to be advantageous during the investigations.

2 Measurement setup

The used measurement setup shown in Figure 2.1 implements the approach of using at least two different relative orientations of object and reference pattern to resolve the ambiguities, which are typical for the deflectometric problem [1]. For realizing this internal distance reference, the setup features a directly driven linear stage of type Standa 8MTL1301-170-LEn1-200, which has a travel of 170 mm with an encoder resolution of 50 nm and a bidirectional repeatability of ± 0.5 µm. As realization of the reference patterns a medical-grade grayscale liquid crystal display of type NEC MD211G5 with a resolution of 2048 x 2560 pixels is used.

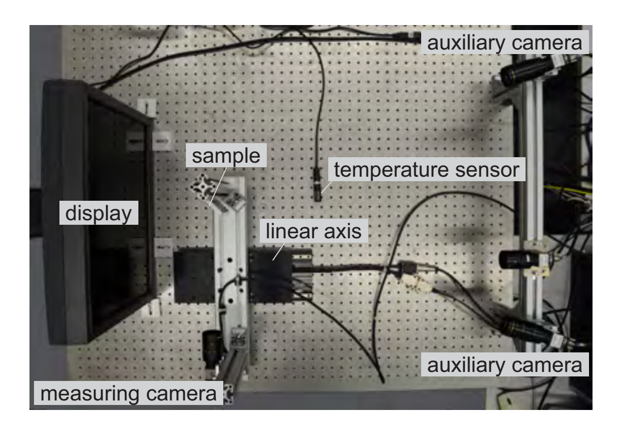


Figure 2.1: Deflectometric measurment setup according to the approach of using two different relative orientations of object and display in order to resolve ambiguities [1].

In contrast to earlier systems used by the authors, for the first time the measuring camera and the test object instead of the display were mounted on the linear stage. As a result, the moving masses are re-

duced and, in addition, the display can be attached very firmly to the optical table with numerous supports, so that the structure has good long-term stability. On the downside, however, it was noticed that different masses of the measurement objects coupled to the linear stage result in very small but detectable tilting movements, which should be addressed in future optimizations.

An essential element of the measurement setup is the stereophotogrammetric auxiliary camera system, which enables an in-situ determination of the display topography as one of the most important deviation influences. The used cameras are all of type IDS UI-3060CP, have a resolution of 2.35 megapixels and are equipped with high-quality lenses. A 35 mm Ricoh FL-BC3518-9M lens is used on the measuring camera, the auxiliary camera system has 16 mm lenses of type Kowa LM16XC.

3 Pattern properties and display calibration

The known phase shifting technique using a heterodyne method for deconvolution of the phase information was subjected to a detailed statistical analysis in order to maximize the robustness on the one hand – i. e. to minimize the probability of deconvolution errors – and on the other hand to increase the statistical deviations of the spatial coding – i. e. minimizing the effects of phase noise. A mathematically well-founded optimality criterion for setting the three wavelengths of the heterodyne method at a given base wavelength was formulated and verified. Compared to conventional approaches, an optimized wavelength ratio can reduce the probability of unwrapping errors by several orders of magnitude [2].

To minimize the statistical deviations of the spatial coding of the display surface by means of phase shifting, it is important to use the most favorable fringe period in each situation. The phase noise model developed by Fischer et al. [3] [4] and the propagation of the deviations into the object space provide an experimental method for determining the optimal wavelength in the respective situation. Figure 3.1 shows that an increase of the wavelength leads to a steady reduction in the phase deviation due to the associated increase in fringe contrast. But considering the physical wavelength

in the display surface, the minimum noise level of the spatial coding is achieved at a comparatively small, clearly definable wavelength. Based on this statistical model, also a strategy to fuse phase measurements with different base wavelengths was implemented, in order to locally achieve the most favorable noise level possible on strongly or unevenly curved object surfaces.

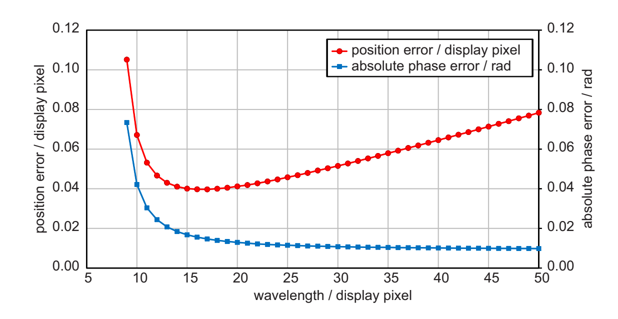


Figure 3.1: Experimentally determined minimization of the statistical position error for a given measurement situation by determining the optimal pattern period.

Since the refraction in the glass substrate of liquid crystal displays represents a significant systematic influence of deviation in deflectometric measurements, special attention was paid to formulating a model describing this influence and an experimental procedure for its determination, which is as precise as possible and at the same time easy to apply. An approach previously followed by the authors has been subjected to extensive revision and refinement. As a result, a partly self-calibrating procedure for determining the refractive properties of the transparent layer was developed, which eliminates or reduces uncertainties of the previous implementation [5] [6]. Following this method, the refractive influence of liquid crystal displays can be characterized and taken into account with previously unattainable accuracy.

Great effort was made to determine the topography of liquid crystal displays. As part of this, various measurement approaches to determine the topography were compared and influences on the topography (gravity, temperature, ...) and temporal variations of the topography were analyzed. As a result, a photogrammetric method

was implemented that enables the determination of the topography directly in the deflectometric setup and, by using the phase coding also used for the deflectometric measurement process, enables a direct determination of the position of each individual display pixel. The auxiliary camera system shown in Figure 2.1 serves primarily for this purpose.

Influences on the optical detection, such as in particular the refraction in the glass substrate, are considered and corrected during the topography measurement. After performing comparative investigations of different approaches, the description of the display topography is currently based on 2D polynomials, which enable the efficient determination of individual surface points of the display and the associated surface normals during the measurement process. The implemented model also enables higher-order deviations to be taken into account, such as the influence of the arc length due to stronger flatness deviations or local deviations in the position of individual pixels. However, it should be noted that these effects, which are typically in the low single-digit micrometer range, cannot be reliably differentiated from noise or higher-frequency residual errors of the camera system due to the large measurement volume (the screen diagonal is more than 540 mm).

4 System calibration

Regarding the system calibration, it was of particular interest to what extent a simultaneous calibration of different components offers advantages or disadvantages compared to a sequential calibration. It was found that simultaneous approaches seem to work well and achieve a high internal accuracy, but that when compared with external reference data, the solutions found this way often show strong deviations, so that the external accuracy or correctness of the calibration is not satisfactory. The behavior observed and supported by simulation calculations indicates that the associated systems of equations converge, but are very susceptible to even small, systematic disturbances due to low condition numbers. A calibration routine is therefore preferred that breaks down the process into individual

sub-steps and, if possible, uses reference information that is not part of the deflectometric arrangement itself.

In a first step a classic photogrammetric calibration with ten parameters for describing the intrinsic orientation of all involved cameras is carried out using calibrated targets. In this step, in order to avoid systematic errors, it is important to ensure that the spectral distribution of the light source that is used to illuminate the targets is as similar as possible to that of the background lighting of the liquid crystal display.

The topography of the liquid crystal screen is then determined directly in the deflectometric setup, but independently of the deflectometric principle by means of a stereophotogrammetric camera system, as shown in Figure 2.1. In this step, external information, in particular about the pixel spacing of the display as obtained from a measurement by means of an optical coordinate measuring machine, is ideally used to increase the accuracy.

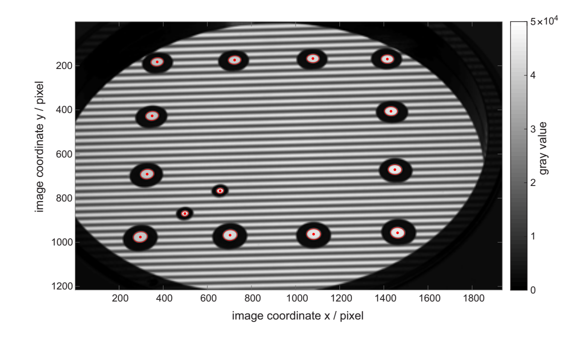


Figure 4.1: Calibration mirror with attached photogrammetric targets. For better visualization, the shown image was composed of two seperate images, one with exposure adjusted to the fringe pattern and one only showing the photogrammetric targets. The red overlay indicates the results from ellipse detection.

To determine the relative orientation of the measuring camera and the (not directly observable) display, a plane mirror as shown in Figure 4.1 with a diameter of 100 mm is used, which is equipped with photogrammetry targets. The relative distances of the targets were measured by an optical coordinate measuring machine and allow the

determination of the mirror position by using only one camera. The mirror can therefore be freely positioned during the calibration process. Ideally, several measurements with different orientations of the mirror are combined, to maximize the robustness and the accuracy. But even in this case it has proven to be disadvantageous to make use of the obvious overdetermination of the problem and to have one or more of the orientation parameters of the calibration mirror be adjust as free parameters.

This sequential procedure has proven to be very flexible and, thanks to the different sources of information, also enables systematic residual errors to be identified, so that these can be further minimized through iterative improvements to components, calibration routines and algorithms.

5 Measurement results

An assessment of the form measurement deviation that can be achieved using the developed approaches and procedures is currently based on the measurement of selected flat and, in particular, spherical test specimens. When selecting the test specimens, it was the declared aim to utilize the usable dynamic range of the created measurement setup as far as possible with regard to the maximum curvature and dimension of the test objects. Using simulation calculation, a convex, spherical test object with an aperture diameter of 50 mm and a radius of curvature of -100 mm was identified as the extreme value, which corresponds to a high aperture ratio of f/1. The other end of the specimen range is formed by a concave specimen with an aperture diameter of also 50 mm and a radius of curvature of 4 inches. Furthermore, one concave and one convex test specimen each with twice the radius of curvature resulting in an aperture ratio of f/2 and a plane mirror, all with an aperture diameter of 50 mm, were used for the qualification measurements.

The measurement data show that this selection of objects can cover unfavorable borderline cases of the measurement setup. Figure 5.1 shows the resulting cases of maximum and minimum used display area. Thus, global effects of the display (shape deviation, viewing angle dependency, ...) as well as local effects (pixel structure, stripe

pattern, ...) impact the measurements as potential sources of deviation.

The absolute radii of curvature were determined by a MarSurf LD 120 combined contour and roughness measurement station. The specified form deviations of the test specimens verified by the contour measurements are $\lambda/4$ in the case of the spherical mirrors and $\lambda/10$ for the plane mirror. It therefore seems justified to ascribe most of the deviations described below to the deflectometric measurement.

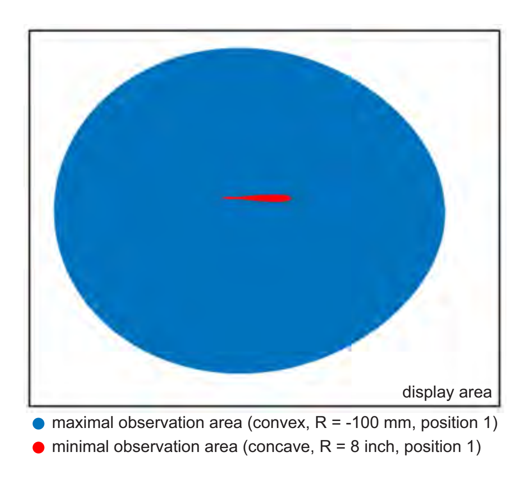


Figure 5.1: Minimal and maximal display observation areas within the range of chosen measurement objects.

Table 1 summarizes the parameters that were achieved on the five test objects in the course of a measurement campaign. It should be emphasized that all measurements were carried out with the system geometry unchanged, i. e. that apart from the sample position no adaptions to the individual properties of the mirrors have been made. For the special case of the plane mirror, only the range of deviations from a best fit plane and the standard deviation of these deviations are listed as parameters. For spherical mirrors, based on guidelines from the field of geometric metrology, a distinction is made between a *probing error size* – in this case the deviation of the radius of the best fit sphere from the reference radius – and a *prob-*

ing error form – here the range of radial deviations from the best fit sphere. In addition, the standard deviation of the radial deviations of this evaluation is given. In order to be able to better assess the influence of the *probing error size* on the *probing error form*, the *probing error total* is also specified, which is the peak-to-valley values of the radial deviations from the spherical surface with the respective reference radius.

Table 1: Measurement deviations achieved on test specimens with different radii of curvature. All mirrors have an aperture diameter of 50 mm. A margin of 1 mm is not taken into account for the evaluation. Further explanation of the parameters in the text.

nominal radius	reference radius (mm)	best fit radius (mm)	probing err. size (µm)	probing err. form (µm)	standard deviation (µm)	probing err. total (µm)
∞ (flat)	∞	–	–	(0.24)	0.06	0.24
8 inch (concave)	203.114	203.138	24.09	0.32	0.05	0.40
4 inch (concave)	101.513	101.502	-11.14	0.61	0.10	0.71
-200 mm (convex)	-200.073	-200.02	52.52	0.47	0.08	0.65
-100 mm (convex)	-99.941	-99.926	14.54	0.92	0.13	1.16

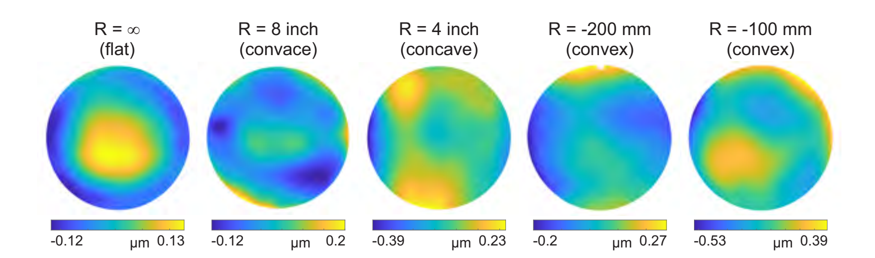


Figure 5.2: Visual representation of the local deviation from the best fit element as taken for probing error form according to Table 1. The color coding of the individual images is selected differently, and each encompasses the range of the corresponding probing error form.

The results in Table 1 show that the *probing error form* for all test objects is less than 1 µm. As stated, these values are the range of all radial deviations. For the standard deviation, here comparable to

the RMS error, values in the two-digit to lower three-digit nanometer range are achieved. The *probing error size* is consistently in the double-digit micrometer range, which however cannot not be interpreted in a quantitative way, since only more or less small sections of a complete spherical surface were measured. The indication of the probing error with respect to a spherical surface of the respective reference radius appears more meaningful, whereby an additional spherical deviation is added to the *probing error form*. As a result, the *probing error total* is on average approx. 25% higher than the *probing error form*.

In Figure 5.2, the radial deviations from the best fit sphere are shown for each of the five measurements listed. The range of these radial deviations corresponds to the *probing error form* from Table 1. The range of the color coding is chosen differently for the five samples and its interval corresponds to the *probing error form*. Due to the spatial frequencies, the local distribution of the residual deviations appears to be systematic in each case, but differ significantly across the test objects.

To minimize the influence of the integration process on the form measurement, existing integration techniques were further developed and compared. An implemented simulation environment has proven to be extremely valuable, as it allowed to use synthetic data to understand higher-order deviations and to minimize their impact. In addition to a local integration approach – which correctly takes into account the imaging distortions of the measuring camera and, compared to earlier work [1], also eliminates curvature-dependent residual errors through an iterative evaluation – two global, model-based approaches were implemented. With these, a polynomial surface – optionally a conventional 2D polynomial or a Zernike polynomial – is adapted to the measured surface normals, whereby any points from internal distance reference can be used for regularization with variable weighting.

In Figure 5.3, for one of the measurements from Figure 5.2 the results obtained from different integration methods are compared. In principle, only the local approach is able to map higher spatial frequencies, but with regard to the form, all three approaches show very good agreement. With the same polynomial order, however, the Zernike polynomials have a somewhat greater tendency to overshoot

at the edge of the test object, which means that the arithmetic range is usually somewhat larger. The results from Table 1 and Figure 5.2 were calculated using 2D polynomials of order 12.

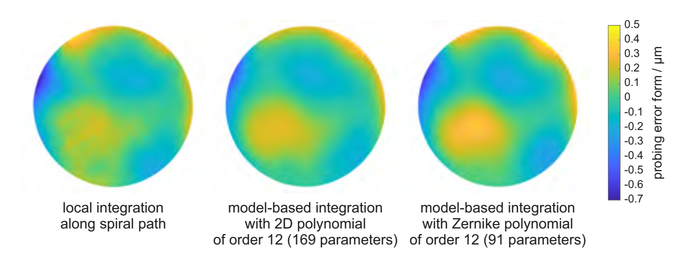


Figure 5.3: Comparison of different integration strategies exemplified by the convex mirror with R_{nom} = -100 mm. The color coding of the individual images is chosen identically and covers the interval [-0.7 µm, 0.5 µm].

6 Summary

A significant improvement of the absolute accuracy of phase measuring deflectometry was achieved by the performed work, which covers a wide range of aspects that contribute to the absolute accuracy. It can be stated that the sub-systems and sub-problems of a deflectometric setup can now be mastered at about the same level as the underlying photogrammetric principle. As a matter of fact, at least some of the remaining deviations can be attributed to not completely corrected optical distortions of the used cameras. These and other systematic residual errors are subject of ongoing investigations.

Acknowledgement

The authors gratefully acknowledge the funding of this work by Deutsche Forschungsgemeinschaft (DFG) under grant Pe1402/6-1.

References

1. M. Petz, “Rasterreflexions-Photogrammetrie – Ein neues Verfahren zur geometrischen Messung spiegelnder Oberflächen,” Dissertation, Technische Universität Braunschweig, 2006, Schriftenreihe des Instituts für Produktionsmesstechnik, Band 1, Aachen: Shaker.
2. M. Petz, H. Dierke, and R. Tutsch, “Wellenlängenoptimierung bei Heterodyn-Phasenschiebeverfahren,” *tm - Technisches Messen (published online ahead of print)*, 22 Sep. 2020.
3. M. Fischer, M. Petz, and R. Tutsch, “Vorhersage des Phasenrauschens in optischen Messsystemen mit strukturierter Beleuchtung,” *tm - Technisches Messen*, vol. 79, no. 10, pp. 451–458, 2012.
4. M. Fischer, M. Petz, and T. R., “Modellbasierte Rauschvorhersage für Streifenprojektionssysteme – Ein Werkzeug zur statistischen Analyse von Auswertealgorithmen,” *tm - Technisches Messen*, vol. 84, no. 2, pp. 111–122, 2017.
5. M. Petz, H. Dierke, and R. Tutsch, “Photogrammetric determination of the refractive properties of liquid crystal displays,” *tm - Technisches Messen*, vol. 86, no. 6, pp. 319–324, 2019.
6. H. Dierke, M. Petz, and R. Tutsch, “Photogrammetrische Bestimmung der Brechungseigenschaften von Flüssigkristallbildschirmen,” in *Forum Bildverarbeitung 2018*, Längle, T. and Puente León, F. and Heizmann, M., Ed. Karlsruhe: KIT Scientific Publishing, 2018, pp. 13–24.

Concept for collision avoidance in machine tools based on geometric simulation and sensor data

David Barton[1], Patrick Männle[1], Sven Odendahl[2], Marc Stautner[2], and Jürgen Fleischer[1]

[1] Karlsruhe Institute of Technology, wbk Institute of Production Science, Kaiserstraße 12, 76131 Karlsruhe, Germany
[2] ModuleWorks GmbH, Henricistraße 50, 52072 Aachen, Germany

Abstract Collisions are a major cause of unplanned downtime in small series manufacturing with machine tools. Existing solutions based on geometric simulation do not cover collisions due to setup errors. Therefore a concept is developed to enable a sensor-based matching of the setup with the simulation, thus detecting discrepancies. Image processing in the spatial and frequency domain is used to compensate for harsh conditions in the machine, including swarf, fluids and suboptimal illumination.

Keywords Manufacturing, collision avoidance, frequency domain

1 Introduction

In order to remain competitive in a global market, manufacturing is under pressure to continuously improve quality, costs and flexibility. There is a trend towards more variety in the final products, leading in turn to smaller batch sizes in production, including single-part production and mass customisation [1]. To be able to produce such parts in an economic way, it is necessary to optimise different stages of the production process. During the preparation phase, the planning and setup effort need to be minimised, which relies

heavily on experience: skilled workers know how to setup production for a new batch and experienced engineers are needed to safely plan the manufacturing process. The scarcity of these skills and the cost of training increase the need for support from digital solutions. New CAM (computer-aided manufacturing) software concepts help to detect problems during process planning, but show deficits when used in an Industry 4.0 environment [2]. The concept of a Digital Twin connects digital process planning with information retrieved via simulation or sensor measurements of the real process [3].

Another approach is to optimise the process on the machine level, for example by reducing downtime through condition-based maintenance and process monitoring [4,5]. One major cause of downtime are collisions between machine parts and production equipment, especially in small series manufacturing [5]. When correctly used, a Digital Twin allows to use advanced CAM algorithms to avoid some collisions already during the planning phase [6]. Other types of collisions cannot be detected in advance due to the potential for human error during the frequent and highly manual operation of setting up fixtures and workpieces [7]. This can be prevented by using a collision avoidance system. To apply such a system, all geometric features need to be modelled correctly by hand or via importing machine geometries and additional elements through given process-planning data. The placement of these elements must be precise to ensure that collision checking and avoidance algorithms work correctly. This is especially the case for the workpiece, fixtures, and other supporting elements, as their geometry or position within the machine can change after the planning stage. To overcome these problems, the present contribution proposes a concept for collision avoidance consisting of a combination of geometric simulation and sensor-based inspection, thus avoiding collisions caused by discrepancies between simulation and real contents of the work area.

2 State of the art

Existing solutions for collision avoidance in machine tools can be divided into the following categories: collision check during process planning, simulation-based dynamic collision avoidance, camera-

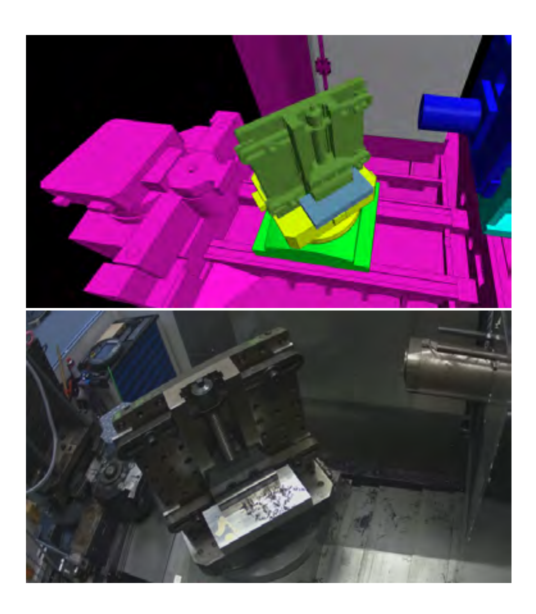


Figure 2.1: Simulation (top) and camera image (bottom) for an exemplary setup.

based monitoring, and monitoring based on distance measurement. Collision check during process planning is a widespread and commercially available approach, in which a geometric model of machine tool, fixture, workpiece, and cutting tool is used to simulate and verify the planned machining steps [8]. In dynamic collision avoidance, a similar geometric model is used to check for collisions (Figure 2.1), however it is integrated in an in-time simulation running during machine operation based on real-time and look-ahead data from the machine control unit [9]. These systems come in two flavours. They either check only for collisions between moving but constant geometries, or they also consider the changing geometry of the workpiece by simulating the material removal in real time as well.

Camera-based monitoring approaches aim to detect discrepancies between the real contents of the machine's work area and a reference geometry (either the geometry used to check the program during process planning, or the situation when previously manufacturing

identical parts). Existing solutions for camera-based monitoring either overlay images from the geometric simulation and the real situation in the machine, and rely on a visual check by the operator [10], or rely on reference images from previous parts of the same type [11]. Monitoring based on distance measurement relies on laser triangulation, ultrasound, or inductive sensors to check the distance between moving parts of the machine (e. g. the main spindle) and obstacles such as the fixture and workpiece [12], however the position and number of sensors is limited due to high costs and limited mounting space.

Another approach to reducing costs due to collisions is collision detection. Acceleration, force or motor current signals are used to detect impacts and unexpectedly high loads, following which the movement of the feed axes is stopped as quickly as possible. This can limit the resulting damage to the machine, thus reducing repair costs and downtime [13]. The approaches described above either require a visual check by the operator, don't cover errors in setting up fixture and workpiece, require significant effort for sensor integration, require reference images from previous manufacturing of identical parts, or aren't able to entirely avoid collisions.

3 General approach

The present approach aims to combine the advantages of simulation and sensor-based approaches in a cost-effective solution for collision avoidance focussing on small-series and single-part manufacturing. In this context, it is especially relevant to ensure the first produced part is a good part, with minimal effort for setting up, running-in and human supervision. The combined system aims to detect mistakes in setting up or in the geometry model as well as discrepancies occurring during manufacturing (e. g. different workpiece shape due to a broken or wrong tool in a previous step, displaced workpiece due to inadequate clamping). The geometry of machine, fixture, workpiece and tool are modelled in the simulation-based collision avoidance system ModuleWorks CAS, and the model is updated during machine operation based on data from the machine control unit and a material removal simulation [9]. The data obtained from the

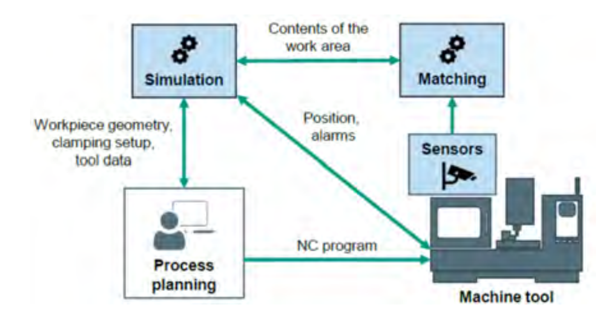


Figure 3.1: System architecture.

control unit comprises all the information necessary to simulate the current and the future state of the machine within a certain time span. Besides pure axis data, this also includes information on states in which the tool should not be allowed to cut material (e. g. during rapid movement or jog movements). If a future collision is detected based on the look-ahead data in the simulation component during automated movement, an alarm is sent to the machine to enable the feed axes to be stopped in time. During manual jog movement, the feed is controlled in such a way that the machine slowly approaches a future collision situation and finally stops before the contact occurs.

In order for CAS to work properly, the setup in the machine needs to be correct at all times. The workpiece in particular must be placed with a high accuracy to allow for safe process conditions. The level of accuracy required depends on the machining process, ranging from below one mm to orders of magnitude smaller. The lower end of this range cannot be checked with contactless sensor data alone within a cost-effective solution. For the placement of fixtures and the workpiece, the system therefore provides the possibility to position objects to a work offset measured by a probing process, which is usually also required to setup the machining process itself. However, the probing process is also prone to collisions because the initial position of the objects still has to be entered manually or based on information from the CAM project. At this stage, but also during the machining process itself, CAS is enhanced by a continuous sensor-based validation of the modelled situation. To accomplish this, the simulation com-

ponent of the collision avoidance system periodically transmits an image of the current geometric model to a separate software system, which is tasked with matching the geometry from the simulation with sensor data acquired in the machine's work area (Figure 2.1). If a discrepancy is detected by the matching algorithm, an alarm is sent to the machine control unit. An overview of the resulting system architecture is shown in Figure 3.1. The approach is tested in the machining centre DMC 60H, though care is taken to develop a solution that is applicable to a wide range of machines. The following section is dedicated to the image processing within the matching algorithm.

4 Image processing

In the first prototypical implementation of the concept, a single camera with a resolution of 1920x1080 pixels is used to observe the machine setup. Simulation-based collision avoidance typically allows for a safety clearance of 3 mm between bodies in the geometric model. In order to detect all critical discrepancies, the measurement and matching in this approach aims to detect deviations of 1 mm or more from the simulated geometry. If required due to the manufacturing process, smaller deviations could then be handled by probing. Damage during probing can be avoided thanks to the previous matching based on camera images.

The aim of image processing within the collision avoidance system is to detect the contours of fixture, workpiece and other obstacles in a sufficient quality for a subsequent comparison with data from the geometric simulation. The conditions in machine tools lead to challenges due to obstruction by swarf (metal chips resulting from the cutting process, ranging from small particles to long tendrils), fluids (oil and coolant), and suboptimal lighting conditions. An example of a workpiece partially covered by swarf and coolant is shown in Figure 2.1. For each of these challenges, suitable image processing methods are evaluated using images acquired in the machining centre DMC 60H.

4.1 Spatial domain

The present approach uses processing in the spatial domain to detect the contours of fixtures and workpieces through Canny edge detection, and to compensate for the influence of lighting and fluids. As no object detection or semantic segmentation has been implemented yet for this application, the region of interest for cropping was selected manually.

The conditions for image acquisition in machine tools can be improved by adding light sources, however the structure of machine tools and the presence of reflecting metallic surfaces mean undesired artefacts due to reflection and shadows remain frequent. Two light sources are used to successively illuminate the scene from different angles. In the resulting images, the position of artefacts linked to the illumination changes. This effect is used by removing edges that do not appear in the same position in both images (within a tolerance of one pixel). The result is shown in Figure 4.1.

Coolant and cutting oil are frequently used to lubricate and cool machining processes. These may cover patches of the workpiece or fixture, thus causing additional edges in the captured images and hampering the detection of contours. The present approach uses the following steps to identify such additional edges:

- Bilateral filter
- Segmentation based on thresholding of pixel colour to identify coolant
- Adding similarly coloured neighbouring pixels to the segment
- Dilation of the identified segment

The original image is subjected to Canny edge detection, then all edges within the identified segment are removed. An example for this procedure is shown in Figure 4.2.

4.2 Frequency domain

Additional image processing is performed in the frequency domain, with the aim of removing edges due to swarf and other causes such

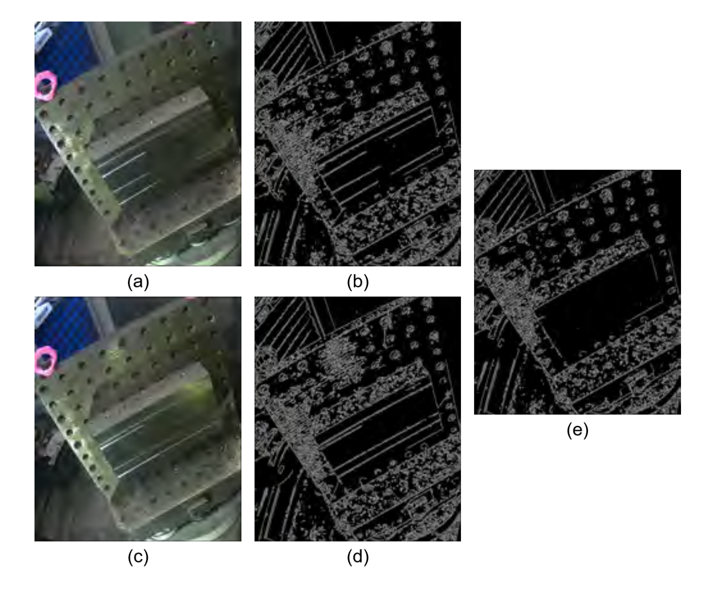


Figure 4.1: Removal of additional edges due to lighting. Images taken while varying the lighting (a, c) display additional edges due to the lighting conditions (b, d). These are identified and removed by comparing the images, thus leading to the improved image (e).

as scratches, chipped painted surfaces, and corrosion. These undesirable features are linked to randomly oriented edges and high spatial frequencies (Figure 4.3).

After using the 2-dimensional discrete Fourier transform (2D DFT) on the original image, the logarithmically scaled amplitude spectrum is subjected to a filter mask. After inverse 2D DFT, Canny edge detection is performed on the filtered image. The filter mask aims to select the dominant directions in an image and eliminate high frequencies, it is generated automatically for each image.

The dominant directions in the image appear as lines in the amplitude spectrum. The spectrum is binarised based on a threshold k, then the number of white pixels is counted for each line passing through the centre of the image, thus creating a histogram of directions. This histogram is smoothed by applying a moving average,

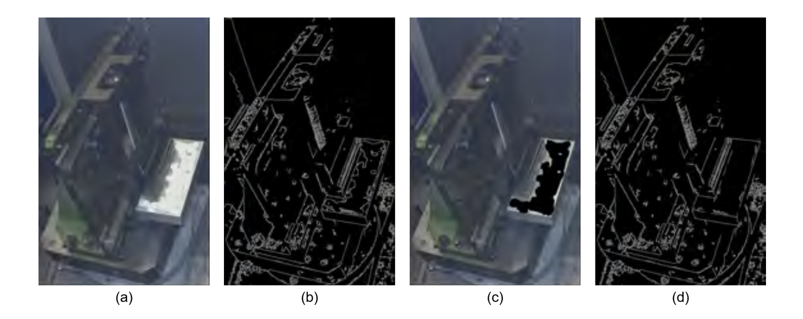


Figure 4.2: Removal of additional edges due to coolant. (a) Original image with coolant; (b) Edges detected in original image; (c) Coolant identified and marked in black; (d) Image after removal of edges due to coolant.

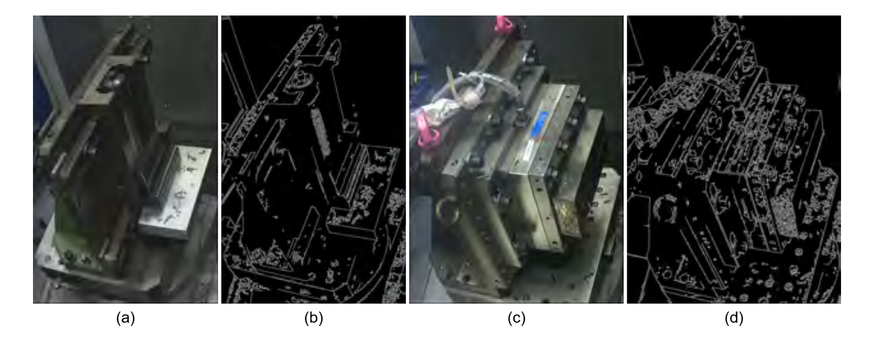


Figure 4.3: Examples of original images containing swarf and edge images without filtering.

then local maxima with a prominence of at least p are determined (Figure 4.4). The filter mask for dominant directions is the union of the following:

- Stripes with a width of b around each of the identified dominant directions,
- A disc with a radius of r_1 in centre of image.

The complete filter mask is the intersection of the above with a low pass filter (with a radius of r_2). Figure 4.5 shows the resulting im-

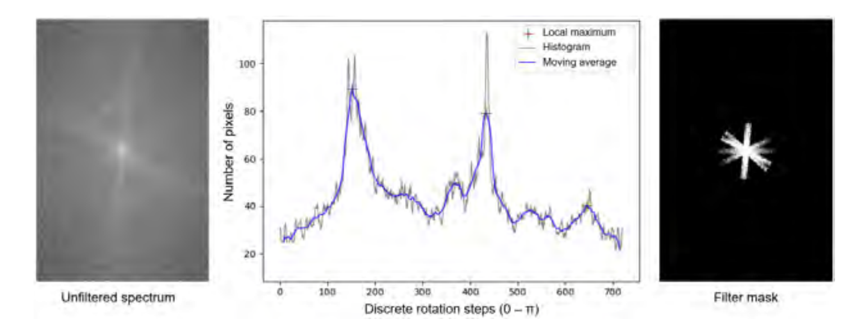


Figure 4.4: Selection of dominant directions in the amplitude spectrum, applied to Figure 4.3a.

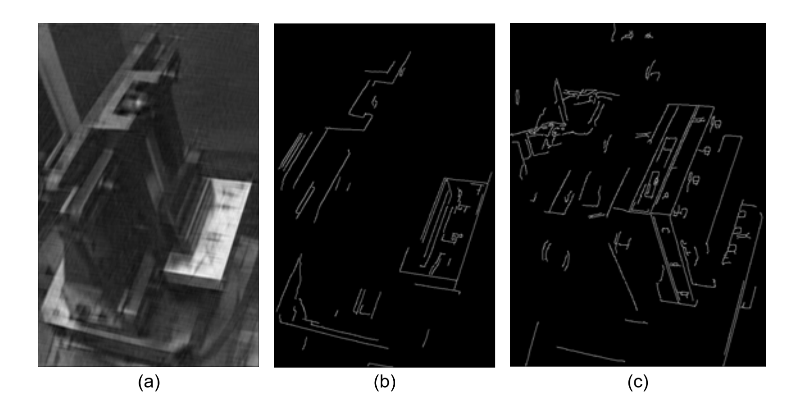


Figure 4.5: Results after filtering. (a) Fig. 4.3a after filtering in frequency domain and inverse transformation; (b) Edges detected in Fig. 4.5a; (c) Edges detected after filtering of Fig. 4.3c.

ages after filtering, inverse DFT and Canny edge detection for the examples from Figure 4.3.

The parameters k, p, $r1$, $r2$, b and the parameters for Canny edge detection are determined manually based on a representative selection of images, whereas the automatically generated filter mask adapts to scenes with different orientations.

5 Summary and further work

A concept was developed for a collision avoidance system covering a larger range of collision causes than existing solutions and especially well-suited to small series and single part manufacturing. The proposed system runs during the operation of a machine tool and combines a state-of-the-art geometric simulation with a sensor-based inspection of the work area. The encouraging initial results presented in this contribution concern the processing of images acquired in the harsh conditions of a machine tool's work area. Further work is needed to perform a wider evaluation for a representative selection of workpieces. The authors also plan to implement automated object detection and extend the concept in order to adjust the simulation model and tool path to the measured reality of the working area.

Acknowledgements

This research and development project is supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) on the basis of a decision by the German Bundestag, within the program "ZIM - Zentrales Innovationsprogramm Mittelstand" (Central Innovation Programme for small and medium-sized enterprises). The author is responsible for the contents of this publication.

References

1. F. Piller and A. Kumar, "Mass customization: Providing custom products and services with mass production efficiency," *Journal of Financial Transformation*, vol. 18, no. 3, pp. 125–131, 2006.
2. E.-M. Jakobs, C. Digmayer, S. Vogelsang, and M. Servos, "Not ready for industry 4.0: Usability of CAx systems," in *Advances in Usability and User Experience*, T. Ahram and C. Falcão, Eds. Cham: Springer International Publishing, 2018, pp. 51–62.
3. B. Schleich, M.-A. Dittrich, T. Clausmeyer, R. Damgrave, J. A. Erkoyuncu, B. Haefner, J. de Lange, D. Plakhotnik, W. Scheidel, and T. Wuest, "Shifting value stream patterns along the product lifecycle with digital twins," *Procedia CIRP*, vol. 86, pp. 3 – 11, 2019, 7th CIRP Global Web Conference – Towards shifted production value stream

patterns through inference of data, models, and technology (CIRPe 2019). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212827120300639

4. D. Barton, P. Gönnheimer, F. Schade, C. Ehrmann, J. Becker, and J. Fleischer, "Modular smart controller for industry 4.0 functions in machine tools," *Procedia CIRP*, vol. 81, pp. 1331–1336, 2019.
5. D. Barton, R. Stamm, S. Mergler, C. Bardenhagen, and J. Fleischer, "Industrie 4.0 Nachrüstkit für Werkzeugmaschinen: Modulare Lösung für zustandsorientierte Instandhaltung und Prozessüberwachung," *WT Werkstattstechnik*, vol. 110, no. 7-8, 2020.
6. D. Plakhotnik, L. Glasmacher, T. Vaneker, Y. Smetanin, M. Stautner, Y. Murtezaoglu, and F. van Houten, "Cam planning for multi-axis laser additive manufacturing considering collisions," *CIRP Annals*, vol. 68, no. 1, pp. 447 – 450, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0007850619300332
7. E. Abele, C. Brecher, S. C. Gsell, A. Hassis, and D. Korff, "Steps towards a protection system for machine tool main spindles against crash-caused damages," *Production Engineering*, vol. 6, no. 6, pp. 631–642, 2012.
8. P. Hehenberger, "Computerunterstützte Fertigung," 2011.
9. M. Armendia, T. Fuertjes, D. Plakhotnik, J. Sossenheimer, and D. Flum, "Cyber-physical system to improve machining process performance," in *Twin-Control*, M. Armendia, M. Ghassempouri, E. Ozturk, and F. Peysson, Eds. Cham: Springer International Publishing, 2019, vol. 58, pp. 197–208.
10. P. Sommer, Ed., *iWindow - Intelligentes Maschinenfenster: Abschlussbericht Verbundforschungsprojekt*, ser. Fortschritt-Berichte VDI Reihe 2, Fertigungstechnik. Düsseldorf: VDI Verlag, 2018, vol. Nr. 697.
11. Heidenhain GmbH, "Optionen und Zubehör für TNC-Steuerungen."
12. T. Li, Y. Wang, K. Liu, H. Liu, J. Zhang, X. Sheng, and D. Guo, "Virtual grid and BPNN based collision avoidance control of automatic fixture system," *The International Journal of Advanced Manufacturing Technology*, vol. 95, no. 5-8, pp. 2843–2853, 2018.
13. T. Rudolf, C. Brecher, and F. Possel-Dölken, "Contact-based collision detection—a new approach to avoid hard collisions in machine tools," in *International Conference on Smart Machining Systems*, 2007.

Lokalisierung von Flammen und Glut für das automatisierte Löschen von Bränden

Fabian Stoller[1], Felix Kümmerlen[2] and Alexander Fay[1]

[1] Helmut-Schmidt-Universität, Institut für Automatisierungstechnik, Holstenhofweg 85, 22043 Hamburg
[2] Wehrwissenschaftliches Institut für Schutztechnologien – ABC-Schutz (WIS), Humboldtstraße 100, 29633 Munster

Zusammenfassung Die Zulassung von Feuerlöschern erfordert die Durchführung von Versuchen für den Nachweis der Löschfähigkeit eines Feuerlöschermodells. Damit dieser Versuch unabhängig von den Fähigkeiten menschlicher Löschmeister wird, soll die Durchführung automatisiert werden. Dafür sollen Algorithmen gefunden werden, die mithilfe einer Farb- und einer LWIR-Kamera Flammen und Glutnester lokalisieren können. Diese Informationen sollen genutzt werden, um den Löschversuch effektiv und effizient durchzuführen. Dafür werden in diesem Beitrag sechs Algorithmen zur Lokalisierung von Flammen in Farbbildern und drei Algorithmen zur Lokalisierung von Glut in den Bildern einer Infrarotkamera anhand der Kriterien Sensitivität, Falsch-Positiv-Rate, Intersection over Union und Ausführungsgeschwindigkeit verglichen, um jeweils einen passenden Algorithmus auszuwählen.

Keywords Lokalisierungsalgorithmen, Feuerlokalisierung, Glutdetektion

1 Einführung

Tragbare Feuerlöscher sind ein zentraler Bestandteil von Brandschutzmaßnahmen an Arbeitsstätten und im öffentlichen Nahverkehr. Die Menge an Feuerlöschern, die bereitzustellen ist, richtet sich

zum einen nach der jeweiligen Brandgefahr und zum anderen nach den Eigenschaften des Feuerlöschers. In der DIN EN 3-7 [1] sind sowohl die Anforderungen an Feuerlöscher als auch deren Zertifizierung beschrieben. Für die Prüfung der Fähigkeit eines tragbaren Feuerlöschers, Brände der Brandklasse A (Feststoffe, die unter Bildung von Glut brennen [2], z. B. Holz) zu löschen, schreibt die angegebene Norm das Löschen eines Testobjekts mit definiertem Aufbau vor. Aktuell wird diese Prüfung von menschlichen Löschmeistern durchgeführt, die auf Basis langjähriger Erfahrung ihre Vorgehensweise beim Löschen des Testbrandes an die begrenzt verfügbare Menge an Löschmittel in einem Feuerlöscher angepasst haben. Damit sie das Feuer effektiv bekämpfen können, differenzieren sie zwischen Flammen und Glut und wenden verschiedene Löschtechniken an, um diese jeweils zu bekämpfen. Mit dem Ziel, die Vergleichbarkeit der Ergebnisse dieser Prüfung und somit die Qualität der Zertifizierung insgesamt zu verbessern, soll der Löschvorgang zur Zertifizierung von Feuerlöschern automatisiert werden. Dafür müssen Flammen und Glut in einem Brand mit geeigneten Bildverarbeitungs-Algorithmen erkannt werden. Auf Basis der mit diesen Algorithmen gewonnenen Informationen über den Brand werden Ziele und Zielbereiche vorgegeben, in denen das Löschmittel aufzubringen ist. Das Ziel dieser Arbeit ist es, Methoden zur Lokalisierung von Flammen und Glut vergleichend zu evaluieren, um so jeweils einen für das automatisierte Löschen eines Normbrandes geeigneten Algorithmus auszuwählen.

Diese Arbeit präsentiert in Abschnitt 2 den Stand der Forschung auf dem Gebiet der Feuerlokalisierung und der Glutlokalisierung. Anschließend werden anhand der Rahmenbedingungen des Normbrandversuchs relevante Kriterien für die Auswahl geeigneter Algorithmen hergeleitet. In Abschnitt 4 wird dann eine Vorauswahl an Algorithmen anhand der gefundenen Kriterien vergleichend bewertet und jeweils ein Algorithmus für die weitere Entwicklung eines Systems für die automatisierte Durchführung der Normbrandversuche ausgewählt.

2 Stand der Forschung

2.1 Lokalisierung von Flammen

Einen guten Überblick über die bis 2013 entwickelten Algorithmen zur Detektion von Feuer und Rauch bietet [3]. Die dort aufgeführten Algorithmen basieren auf einer Kombination mehrerer charakteristischer Eigenschaften von Flammen. Besonders relevant in diesem Zusammenhang ist die Farbe einer Flamme, die als häufigstes Kriterium zum Beispiel in [4–6] verwendet wird. Zusätzlich verwenden zum Beispiel die Algorithmen in [7–9] die zeitliche Veränderung der Form von Flammen oder die Unregelmäßigkeit ihrer Bewegung wie etwa in [10]. Diese unterscheidet sie von den meisten anderen beweglichen Objekten, da diese zumeist regelmäßige Bewegungsmuster haben. Für eine Lokalisierung hingegen wird die charakteristische Bewegung der Flamme kaum eingesetzt.

Die Algorithmen zur Lokalisierung von Feuer in Kamerabildern in [8, 11, 12] setzen verschiedene Formen farbbasierter Features ein. In [12] wird jedes Pixel mit einem naiven Bayes Klassifikator entweder der Klasse *Feuer* oder *kein Feuer* zugeordnet. Zusätzlich wird jedes Bild in Superpixel unterteilt, und diese werden anhand ihrer jeweiligen Texturen den genannten Klassen zugeordnet. In [11] werden die Pixel jedes Bildes anhand der Auftrittswahrscheinlichkeiten bestimmter Farbwerte in Flammen der Klasse *Feuer* zugeordnet. Dafür werden anhand von Trainingsdaten diese Auftrittswahrscheinlichkeiten bestimmt. Anschließend wird anhand der Entropie eine weitere Eigenschaft von Flammen überprüft, um die Fehldetektionsrate zu verringern. Die in [8] für die Lokalisierung von Deflagrationen vorgestellte Methode setzt neben einem regelbasierten Kriterium für die Farben von Flammen und einem Kriterium für die Ausdehnung der als Flammen segmentierten Bereiche ein Hintergrundmodell ein.

Einige auf CNN basierenden Methoden setzen mit großen Bilddatensätzen vortrainierte CNNs ein, welche anhand vergleichsweise kleiner Datensätze mit domänenspezifischen Bildern auf die Lokalisierung von Flammen angepasst werden. Diese Vorgehensweise wird als Transferlernen bezeichnet. Ein solcher Ansatz wird zum Beispiel in [13] vorgestellt. Dort wird ein auf SqueezeNet [14] basierendes

CNN mit Hilfe von Bildern von Feuer auf die Flammenerkennung spezialisiert. Für die Lokalisierung wird eine Featuremap aus dem CNN als Maske verwendet. In [15] werden mehrere CNN für die Lokalisierung von Objekten mit Transferlernen auf Feuer spezialisiert. Von den dort vorgestellten Architekturen hat YOLO [16] die besten Resultate hervorgebracht. In [17] wird DeepLabv3 [18], ein CNN für die semantische Segmentierung von Feuer, mittels Transferlernen angepasst. Dieser Ansatz verspricht die Flammen am genausten zu lokalisieren, ist aber zugleich auch der komplexeste Ansatz.

2.2 Lokalisierung von Glut

Die bildverarbeitungsbasierte Lokalisierung von Glut ist bisher kaum als dezidiertes Problem erforscht. Es existieren jedoch verschiedene Ansätze für sehr ähnliche Probleme. Ein Einsatzgebiet ist zum Beispiel die Lokalisierung von schwelenden Torfbränden mit Erdbeobachtungssatelliten. In [19] werden anhand der Daten eines Infrarotspektrometers die Bildbereiche ausgewählt, die Temperaturen im für schwelende Torfbrände typischen Bereich aufweisen. Dieser liegt deutlich unterhalb der Temperaturen von mit Flammen brennenden Bereichen und deutlich oberhalb der Umgebungstemperatur. Der relevante Temperaturbereich für Torfbrände unterscheidet sich von dem für glühendes Holz, die Lokalisierung kann aber auf die gleiche Art durchgeführt werden.

In [20] werden zwar keine Glutnester detektiert, allerdings lässt sich die Methode, die hier zur Detektion von Hotspots auf Photovoltaik-Anlagen eingesetzt wird, ebenso für die Detektion von heißen Stellen in einem gelöschten Brand einsetzen, also zur Identifikation von Glutnestern. Dabei setzen die Autoren auf den Einsatz von k-Means-Clustering, um Bereiche mit von der Umgebung stark abweichenden Temperaturen zu finden. Mit dieser Methode wird der große Temperaturunterschied zwischen Umgebung und dem Bereich von Interesse für eine Lokalisierung genutzt. Dieser Methode fehlt aber eine Berücksichtigung der absoluten Temperatur, sodass zu jedem Zeitpunkt zwei Temperaturcluster gesucht werden. Eine vergleichbare Vorgehensweise findet sich auf Kohlehalden, wo Schwelbrände lokalisiert werden sollen, die bei der Selbsterwärmung der Kohle entstehen können [21].

In [22] werden Brände mit einem Infrarotstereokamerapaar lokalisiert. Für die Identifikation von Pixeln wird ein Schwellwert für die Temperatur von $T = 300°\mathrm{C}$ festgelegt, da Brände eine deutlich höhere Temperatur besitzen als der Hintergrund der Szene. Zusätzlich wurde die Annahme getroffen, dass es sich bei dem Feuer um den größten segmentierten Bereich in den Aufnahmen handelt. Da diese Schwelle jedoch sehr niedrig gewählt ist, werden so nicht nur Flammen, sondern auch Glutnester segmentiert.

3 Anforderungen an die Algorithmen zur Lokalisierung von Flammen und Glut

Beim Normbrandversuch nach DIN EN 3-7 [1] gilt es, mit der vorhandenen Menge an Löschmittel einen möglichst großen Löscheffekt zu erzielen. Dafür muss das Löschmittel so appliziert werden, dass es dort wirkt, wo die Verbrennungsreaktion am intensivsten stattfindet. Diese Stellen sind zum einen die Flammen und zum anderen, sobald die Flammen gelöscht sind, die Glutnester. Eine Lokalisierung von Flammen und Glut soll mittels eines multimodalen Kameraverbunds ermöglicht werden. Darin soll eine Farbkamera für die Lokalisierung der Flammen eingesetzt werden und eine Infrarotkamera für die Lokalisierung der Glutnester. Diese Aufteilung ist gewählt, da so die Möglichkeit besteht, beide Merkmale zu lokalisieren und zu unterscheiden. Die Unterscheidung ist erforderlich, um den Einsatz von Löschmittel entsprechend anzupassen und für die weniger heißen Glutnester auch entsprechend weniger Löschmittel einzusetzen.

Eine geeignete Methode lokalisiert die Flammen, bzw. die Glutnester, möglichst genau, um einen genauen Auftrag des Löschmittels auf die lokalisierten Stellen zu ermöglichen. Für die Bewertung dieses Kriteriums wird zum einen die Wahrscheinlichkeit bestimmt, dass der jeweils getestete Algorithmus eine Flamme erkennt. Dafür wird die Kenngröße der Sensitivität verwendet, deren Berechnung in Gleichung 3.1 beschrieben ist. Sie ist der Quotient aus richtig positiven

(RP) Detektionen und der Summe dieser mit den falsch negativen Detektionen (FN).

$$s = \frac{RP}{FN + RP} \tag{3.1}$$

Zum anderen wird für die korrekt erkannten Flammen bestimmt, wie genau die Lokalisierung ist. Dafür werden die Bounding Boxen der lokalisierten Flammen mit den in den Grundwahrheiten hinterlegten Bounding Boxen verglichen. Die Intersection over Union (IOU) ist der in Gleichung 3.2 dargestellte Quotient aus der Schnittfläche A_s und der vereinigten Fläche A_v von der Bounding Box der Lokalisierung und der Bounding Box der Grundwhrheiten.

$$IOU = \frac{A_s}{A_v} \tag{3.2}$$

Weiterhin soll die Rate an Fehlalarmen möglichst gering sein, damit möglichst kein Löschmittel für falsche Ziele verschwendet wird. Als Maß für die Fehlalarmrate wird die Falsch-Positiv-Rate (FPR) verwendet, welche sich, wie in Gleichung 3.3 dargestellt, als Quotient der falsch positiven (FP) Detektionen und der Summe der FP mit den richtig negativen (RN) Detektionen berechnet.

$$FPR = \frac{FP}{RN + FP} \tag{3.3}$$

Ein letztes Kriterium ist die Ausführungsgeschwindigkeit der Algorithmen. Diese sollte möglichst hoch sein, um die Steuerung des Löschvorgangs in Echtzeit zu ermöglichen und auf Veränderungen der Situation während des Brandversuchs reagieren zu können. Die Entscheidung für jeweils einen Algorithmus zur Lokalisierung von Flammen und Glut wird anhand der oben beschriebenen Kriterien vorgenommen.

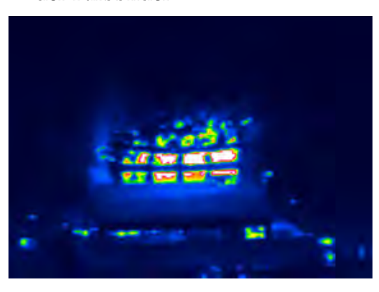
4 Auswertung

Für die Auswertung der vorgestellten Methoden zur Flammenlokalisierung werden 65 Videos aus [12] und [23] sowie eigene

Aufnahmen verwendet, die insgesamt aus 4394 einzelnen Frames bestehen. Die Infrarotbilder stammen aus einem eigenen Datensatz, der 86 Aufnahmen enthält, von denen 43 glühendes Holz zeigen. In Abb. 4.1 sind Beispiele der für die Auswertung verwendeten Farbbilder (Abb. 4.1(a)) und IR-Bilder (Abb. 4.1(b)) dargestellt. Die Implementierung und Auswertung der Methoden erfolgte in Matlab auf einem PC mit Intel Core i7-8565U mit $1,8$GHz Basistakt und 16GB Arbeitsspeicher.



(a) Beispielbild aus dem Testdatensatz der Farbbilder



(b) Beispielbild aus dem Testdatensatz der Infrarotbilder

Abbildung 4.1: Beispielbilder aus einem nach DIN EN 3-7 aufgebauten Brand

4.1 Vergleich der Methoden zur Flammenlokalisierung

Wie bereits in Abschnitt 2.1 dargelegt, gibt es für die Lokalisierung von Flammen in Kamerabildern verschiedene Ansätze. Es werden hierfür die Ansätze aus [8, 11–13, 15, 17] implementiert und anhand der in Abschnitt 3 hergeleiteten Kriterien auf ihre Eignung für den Einsatz beim automatisierten Löschen von Normbränden hin untersucht.

Für das Training von [11] und [12] wird der Datensatz mit Ausschnitten aus Flammenbildern von [12] verwendet. Für das Training der Methoden [13,15,17] werden Bilder aus den Datensätzen [12,23] sowie aus eigenen Aufnahmen verwendet.

In Tabelle 1 sind die Ergebnisse der Auswertung der vorgestellten Algorithmen dargestellt. Auf Basis dieser Ergebnisse lässt sich erkennen, dass die Sensitivität der Erkennung von Flammen bei den Algorithmen [8, 11, 12] am höchsten ist. Eine nur geringfügig niedrigere Sensitivität weisen [13,15] auf. Der Algorithmus nach [17] weist auf dem Testdatensatz die mit Abstand niedrigste Sensitivität auf. Die Auswertung der IOU zeigt, dass die Lokalisierungen des Algorithmus [12] die höchste Übereinstimmung mit den Grundwahrheiten besitzen und die Algorithmen nach [8, 17] die niedrigsten. Diese beiden Algorithmen weisen zusätzlich jeweils eine hohe FPR auf. Die FPR ist insbesondere deshalb so hoch, da in jedem Testbild beliebig viele Fehler passieren können. Bei [13] führen zum Beispiel fehlerhaft segmentierte Bereiche in Bildern, in denen auch eine korrekte Detektion gefunden wird, zu der hohen FPR. Gute Ergebnisse für dieses Kriterium erreichen besonders die Algorithmen nach [11, 15], die im Vergleich mit den übrigen Algorithmen deutlich niedrigere FPR aufweisen. Die Auswertung der mittleren Ausführungszeit ergibt, dass die Ausführung des auf DeepLab basierenden Algorithmus [17] um ein Vielfaches langsamer ist als die der übrigen Algorithmen. Das liegt wiederum an der hohen Komplexität der verwendeten CNN-Architektur. Am schnellsten ist die Ausführung eines Durchlaufs des Algorithmus nach [8], welcher darauf spezialisiert ist. BOWFire [12] und der auf SqueezeNet basierende Algorithmus [13] sind in etwa gleichauf, genauso wie die noch etwas langsameren Algorithmen nach [11] und [15]. Dabei ist anzumerken, dass die Ausführungszeit der auf CNN basierenden Methoden weniger stark von der Größe

der Eingangsbilder abhängt als die übrigen Algorithmen, da die Eingangsschicht der CNN jeweils eine konstante Dimension besitzt.

Tabelle 1: Ergebnisse der Flammenlokalisierungsmethoden auf den Testdaten

Algorithmus	Sensitivität	FPR	IOU	Zeit
[8]	82,12%	93,04%	26,13%	8,24ms
[11]	83,59%	18,63%	53,12%	72,75ms
[12]	81,95%	69,27%	60,08%	30,46ms
[13]	72,53%	69,73%	51,46%	20,71ms
[15]	74,85%	1,86%	45,22%	81,87ms
[17]	43,81%	92,02%	14,78%	583,6ms

Aus der Kombination der Ergebnisse folgt, dass als Algorithmus für das automatisierte Löschen eines Normbrandversuchs nach DIN EN 3-7 [1] die zwei Algorithmen nach [11, 15] in Frage kommen. Dabei weist der Algorithmus nach [11] sowohl eine höhere Sensitivität bei der Erkennung von Flammen als auch eine höhere Genauigkeit in der Lokalisierung sowie eine geringfügig schnellere Ausführungsgeschwindigkeit auf. Der Algorithmus nach [15] hingegen hat hingegen die mit Abstand geringste FPR. Da [11] anhand der Ergebnisse die insgesamt bessere Lokalisierung von Flammen verspricht, wird dieser Algorithmus für den Einsatz im automatisierten Löschen ausgewählt und die etwas höhere FPR akzeptiert.

4.2 Vergleich von Methoden zur Glutdetektion

Die Detektion von Glutnestern und heißen Stellen, die zu einer Wiederentzündung des Brandes führen können, basiert ausschließlich auf dem Temperaturunterschied dieser Bereiche im Vergleich zum Hintergrund. Das Beispielbild in Abb. 4.1(b) zeigt, dass sich diese Bereiche deutlich vom Hintergrund abheben. Im Folgenden werden die Methoden auf Basis von [19,20,22] vergleichend evaluiert. Dafür werden die bereits für die Flammendetektion beschriebenen Kriterien verwendet. Die Ergebnisse werden mit einem eigenen Datensatz aus 86 IR-Bildern von glühendem Holz sowie IR-Bildern ohne Glut generiert.

In Tabelle 2 sind die Ergebnisse der Anwendung der beschriebenen Methoden auf die Testdaten dargestellt. Es ist ersichtlich, dass

Tabelle 2: Ergebnisse der Glutlokalisierungsmethoden auf den Testdaten

Algorithmus	Sensitivität	FPR	IOU	Zeit
[19]	99,15%	41,89%	71,62%	0,1ms
[20]	99,15%	91,76%	64,58%	8,35ms
[22]	99,15%	6,52%	81,04%	0,1ms

alle drei getesteten Algorithmen eine sehr hohe Sensitivität für die Detektion von Glut besitzen. Anhand der mittleren IOU der drei Algorithmen ergibt sich, dass [22] die höchste Übereinstimmung mit den Grundwahrheitswerten aufweist. Die Betrachtung der FPR zeigt jedoch, dass der Algorithmus nach [20] hier am schlechtesten abschneidet, da er aufgrund eines fehlenden Bezugs zu einer absoluten Temperatur eine große Zahl an falsch positiven Detektionen in den Bildern ohne Glut generiert. Der Algorithmus nach [22] besitzt die niedrigste FPR.

Bei der Ausführungsdauer ist die Lokalisierung mit K-Means [20] im Mittel erheblich langsamer als die anderen beiden Methoden. Verglichen mit den Methoden zur Flammenlokalisierung ist diese Methode jedoch sehr schnell, was neben der geringeren Komplexität der Methode auch an den geringeren Auflösungen der Eingangsbilder liegt.

Für die Detektion von Glut ergibt sich der Algorithmus nach [22] als in allen betrachteten Kriterien führend und wird dementsprechend für die Umsetzung des automatisierten Normbrandversuchs eingesetzt, um Glutnester und heiße Stellen nach dem Ablöschen der Flammen in einem Brand zu lokalisieren.

5 Zusammenfassung

Die Automatisierung des Normbrandversuchs nach DIN EN 3-7 [1] erfordert Bildverarbeitungsalgorithmen, die es ermöglichen, Flammen und Glut zu lokalisieren, um den Brand möglichst effizient bekämpfen zu können. Mehrere Algorithmen aus dem Stand der Forschung sind anhand ihrer Sensitivität, Falsch-Positiv-Rate, IOU und Ausführungsgeschwindigkeit verglichen worden, um den für diese Aufgabe am besten geeigneten Algorithmus zu bestimmen.

Für die Flammenlokalisierung eignen sich besonders die Algorithmen [11, 15]. Die IOU und die Sensitivität sind bei [11] höher und auch die durchschnittliche Ausführungsgeschwindigkeit ist etwas geringer als bei dem Algorithmus nach [15]. Mit [15] ist die FPR dagegen deutlich geringer als bei [11]. Daraus resultiert zusammengenommen die Auswahl von [11] für die Lokalisierung von Flammen im Kontext des Normbrandversuchs.

Die Detektion von Glut bietet keinen vergleichbar breiten Stand der Forschung, da die Lokalisierung selten als eigenständiges Problem bearbeitet wird. Die getesteten Algorithmen basieren auf der Segmentierung anhand von Temperaturschwellen und sind aufgrund ihrer geringen Komplexität effizient zu berechnen. Für den Einsatz im Normbrandversuch wird entsprechend der Methode in [22] ein einfacher Schwellwert gewählt, um die Glut zu lokalisieren, da diese sowohl bezüglich der IOU als auch in der FPR und der Ausführungszeit die besten Ergebnisse liefert.

Mit diesen beiden Algorithmen ist die Basis für eine Steuerung des Löschvorgangs im Normbrandversuch gelegt. Die Lokalisierungsergebnisse, die die beiden hier ausgewählten Algorithmen produzieren, lassen in einem Brand die für das effiziente Löschen essentiellen Flammen und Glutnester lokalisieren und als Ziele für das Löschmittel festlegen.

Literatur

1. "Portable fire extinguishers - Part 7: Characteristics, performance requirements and test methods," 2007.
2. Deutsches Institut für Normung and Europäisches Komitee Für Normung, "Brandklassen: Deutsche Fassung EN 2:1992 + A1:2004," 1992.
3. A. E. Çetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y. H. Habiboğlu, B. U. Töreyin, and S. Verstockt, "Video fire detection – Review," *Digital Signal Processing*, vol. 23, no. 6, pp. 1827–1843, 2013.
4. T. Çelik, H. Özkaramanli, and H. Demirel, "Fire and smoke detection without sensors: Image processing based approach," in *15th European Signal Processing Conference*, IEEE, Ed., 2007, pp. 1794–1798.
5. P. Barmpoutis, K. Dimitropoulos, K. Kaza, and N. Grammalidis, "Fire Detection from Images Using Faster R-CNN and Multidimensional

Texture Analysis," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 12.05.2019 - 17.05.2019, pp. 8301–8305.

6. Z. Zhong, M. Wang, Y. Shi, and W. Gao, "A convolutional neural network-based flame detection method in video sequence," *Signal, Image and Video Processing*, vol. 12, no. 8, pp. 1619–1627, 2018.
7. B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, and A. E. Çetin, "Computer vision based method for real-time fire and flame detection," *Pattern Recognition Letters*, vol. 27, no. 1, pp. 49–58, 2006.
8. J. Krooß, F. Kümmerlen, and A. Fay, "Schnelle und präzise Segmentierung von beweglichen und morphologisch variablen Objekten am Beispiel der Deflagrationsdetektion," in *Forum Bildverarbeitung 2018*, F. Puente León, M. Heinzmann, and T. Längle, Eds. Karlsruhe: KIT Scientific Publishing, 2018, pp. 241–252.
9. P. Foggia, A. Saggese, and M. Vento, "Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 9, pp. 1545–1556, 2015.
10. S. Verstockt, "Multi-modal Video Analysis for Early Fire Detection," Dissertation, Universiteit Gent, Gent, 14.12.2011.
11. B. M. N. de Souza and J. Facon, "A fire color mapping-based segmentation: Fire pixel segmentation approach," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 29.11.2016 - 02.12.2016, pp. 1–8.
12. D. Y. T. Chino, L. P. S. Avalhais, J. F. Rodrigues, JR., and A. J. M. Traina, "BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis," pp. 95–102, 2015.
13. K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1419–1434, 2019.
14. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and$<$ 0.5 MB model size," *arXiv preprint arXiv:1602.07360*, 2016.
15. P. Li and W. Zhao, "Image fire detection algorithms based on convolutional neural networks," *Case Studies in Thermal Engineering*, p. 100625, 2020.
16. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement."

17. J. Mlích, K. Koplík, M. Hradiš, and P. Zemčík, “Fire Segmentation in Still Images,” in *Advanced Concepts for Intelligent Vision Systems*, ser. Lecture Notes in Computer Science, J. Blanc-Talon, P. Delmas, W. Philips, D. Popescu, and P. Scheunders, Eds. Cham, Switzerland: Springer International Publishing, 2020, vol. 12002, pp. 27–37.
18. L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation.” [Online]. Available: http://arxiv.org/pdf/1706.05587v3
19. C. D. Elvidge, M. Zhizhin, F.-C. Hsu, K. Baugh, M. R. Khomarudin, Y. Vetrita, P. Sofan, Suwarsono, and D. Hilman, “Long-wave infrared identification of smoldering peat fires in Indonesia with nighttime Landsat data,” *Environmental Research Letters*, vol. 10, no. 6, 2015.
20. A. M. Salazar and E. Q. B. Macabebe, “Hotspots Detection in Photovoltaic Modules Using Infrared Thermography,” vol. 70, p. 10015, 2016.
21. V. Fierro, J. Miranda, C. Romero, J. Andrés, A. Pierrot, E. Gómez-Landesa, A. Arriaga, and D. Schmal, “Use of infrared thermography for the evaluation of heat losses during coal storage,” *Fuel Processing Technology*, vol. 60, no. 3, pp. 213–229, 1999.
22. J. McNeil, “Autonomous Fire Suppression Using Feedback Control for a Firefighting Robot,” Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, 14.12.2015.
23. M. T. Cazzolato, L. P. S. Avalhais, D. Y. T. Chino, J. S. Ramos, J. A. de Souza, J. F. Rodrigues-Jr, and A. J. Traina, “Fismo: A compilation of datasets from emergency situations for fire and smoke analysis,” *Proc. Satell. events*, 2017.

Methods for the localization of supporting slats of laser cutting machines in single images

Frederick Struckmeier[1,2], Philipp Blättner[2], and Fernando Puente León †[2]

[1] TRUMPF Werkzeugmaschinen GmbH + Co. KG
Johann-Maus-Str. 2, 71254 Ditzingen, Germany
[2] Karlsruhe Institute of Technology, Institute of Industrial Information Technology, Hertzstraße 16, 76187 Karlsruhe, Germany

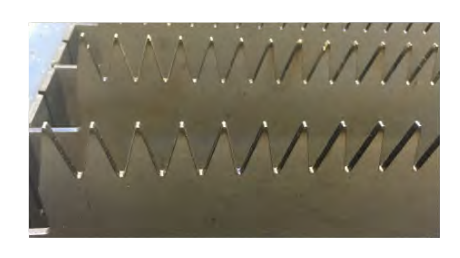
Abstract The supporting slats of laser flatbed machines cause process reliability problems, such as tilted parts colliding with the cutting head. In order to mitigate these problems the position of the supporting points for a part to be cut must be known, before the machines numerical control program can be changed accordingly. Being able to detect the position of supporting slats accurately is necessary to do that. This work compares image processing methods to localize the supporting slats in single images. The best features are based on filters in the frequency domain and can have accuracies above 96 %.

Keywords Image processing, object detection, object localization, laser cutting, laser flatbed machine

1 Introduction

Laser flatbed cutting machines (LFMs) are an important part in the sheet metal production process, as they are able to efficiently cut contours of any form. In the LFM layout, the metal sheet is still, while the cutting head moves above it. Supporting slats are used to support the metal sheet during the cutting process. The slats are metal strips, that are a few millimeters wide and are basically a row

of isosceles triangles (see Fig. 1.1 left). The slats are attached to the pallet at certain positions, where the slat is pushed into a socket.



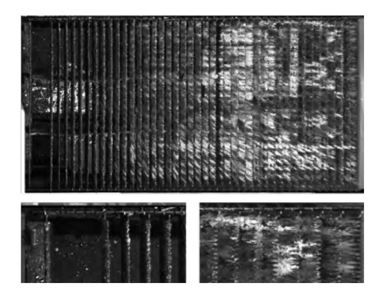


Figure 1.1: Left: Supporting slats of a LFM. Right: An example image of the empty pallet, with detailed sections of the left and right upper corners below.

Whilst being cost efficient and decently robust regarding the adverse conditions under the sheet being cut, the supporting slats cause some problems with the process reliability [1]. For example, a part may tilt after being cut free, depending on the position of supporting tips under that part and the gas pressure. A tilted part can cause collisions with the cutting head leading to downtime of the machine. Also, if cutting right above a tip, the slat might be damaged unnecessarily and the part can have lower quality due to visible marks [1].

In order to prevent these problems, adjustments to the numerical control program of the machine have been suggested, namely changes to the nesting layout [1] and the tool path [2]. However, these approaches assume that the position of the supporting tips relative to the raw metal sheet is known in advance. This is generally not the case with the LFMs in use today.

Reference [3] presented different methods to measure the supporting slats of LFMs. Whilst that work focused on laser triangulation, it pointed out a big advantage of detecting slats in single images: there is almost no auxiliary process time needed. As it is unclear which method is best suited to detect slats in an image, this work compares different methods.

The task is to find an estimator that is first able to identify the slat tips in the image and then translate this information to the slat socket positions of the pallet. A major difficulty is the possible varia-

tion in the appearance of the slats. They can be made from different materials, mostly mild steel, stainless steel and copper. Also, cutting causes wear and tear of the slats. Firstly, the drops of molten material exiting from the cutting kerf stick to the slats. A well-used slat therefore has multiple colours (see Fig. 1.1 right). A different cause of the variation in the images is the background. The machine has two pallets, so one might be above the other. The lower pallet can be seen through the upper pallet from the camera perspective. Also, in an industrial setting there are often scrap pieces of metal on the floor below the pallet, which can also be seen through the pallet.

In the next section we present the state of the art of object detection and localization in single images. In Section 3, the different features and classifiers of this work are explained in detail. The test set and the results of the methods are presented in Section 4, before Section 5 concludes the work.

2 State of the art

Whilst there is a comprehensive body of literature on object detection and localization in images, the problem of localizing supporting slats of LFMs in single images has never been studied before to the knowledge of the authors.

The definition of the terms object detection and object localization in different image processing works is not always the same. Sometimes these terms are used almost interchangeably, because predicting that an object is present in an image is usually based on features, whose appearance in the image can be restricted to certain locations of the image. For the rest of this work we will define object localization to include the detection and accurate estimation of the location of the searched object [4].

An established method for object localization is template matching. A known template is convolved with the image, resulting in the feature image. When detecting a single instance of an object in an image, classification is performed by selecting the highest peak [5].

Another approach to localization of an object in an image is parallel projection. In 2D images, the parallel projection is equivalent to summing the pixel values in a given direction or an integral over

that axis. It is used in some medical image processing works to localize a tool, e. g. a needle, in a 3D image obtained by ultrasound imaging [6].

Most other successful frameworks do not focus on object detection and localization in the sense of this paper. The SIFT algorithm [7] for example can find different known objects at different scales and rotations. However, in our case there is only one object of the same size and problems arise because of high variance in lighting, background and wear conditions.

The most recent and for many use cases very successful method is applying convolutional neural nets for image processing. Because there are only about 200 images available for training and validating such a framework, this approach is not further pursued.

3 Methods

3.1 Image Acquisition and Perspective Transformation

The camera taking the images is mounted on top of the LFM overlooking the pallet outside of the machine body (see Fig. 3.1). The perspective requires a camera with a wide-angle lens.



Figure 3.1: The position of the camera on the machine body.

The image perspective is transformed, so that it displays the scene from a birds-eye view. The resulting image (see Fig. 1.1 right) has a size of 1600 by 3100 pixels. Note that the perspective transformation leads to the slats close to the machine body being seen from above, whereas the slats on the other side of the pallet are seen at an angle.

3.2 Features

Intuitively, slats are vertical lines in the images. The triangular shape of the tips leads to many edges and corners along the slat. They can also be seen as a texture. Hence, the features selected for further study are different edge and corner detectors, Laws' energy measures and a hand-crafted model of slats in the spatial frequency domain. In order to establish a baseline, the unmodified images are also used as an input to the two classifiers introduced in Section 3.3.

Edge and Corner Detectors

An edge in image processing is simply put a large change in brightness along a line in the image. The change in brightness can be detected by analyzing the first or second derivative of pixel values. Hence, two approaches are tested, namely the gradient-of-Gaussian filter and the Difference-of-Gaussians (DoG) filter.

The first is an approximation based on the gradient of the image. It can be shown that a smoothing of the image with a Gaussian low-pass filter and a differentiation of the image is equal to a convolution of the image with the derivative of a Gaussian low-pass filter [5]. Discrete sampling of the Gaussian low-pass filter will result in a discrete formulation of the gradient-of-Gaussian filter.f

The Laplacian-of-Gaussian filter is an approximation of the second derivative of the smoothed image [8], which can be used for edge detection. The Laplacian filter is the simplest approximation of the second derivative obtained by a convolution. However, it is rather sensitive to noise, which is why the image is smoothed with a Gaussian low-pass filter. A discrete implementation is approximated by the DoG filter, which is based on the difference of two Gaussian functions with different standard deviations [5].

For corner detection, the Harris corner detector is used [9]. The idea for this detector is to take a small window of the image and test how much change happens to the values if the window is shifted by a small distance in all directions. In an area with no edges or corners, the change in pixel values will be low. If an edge is present, the change will be low in direction of the edge. At a corner however, there is significant change in all directions, when the window

is moved. A more formal and detailed description can be found in [5,9].

Laws' Energy Measures

Another approach is to interpret the slats as textures. One of the most used texture processing algorithms are Laws' energy measures [10]. They consist of a set of quadratic matrices of variable size. The matrices of size 5×5 were chosen, as they were shown to be a good trade-off between information content and computational speed [10]. The matrices are defined as the result of all combinations of outer products of four vectors, representing the detection of levels ($\mathbf{l_5}$), spots ($\mathbf{s_5}$), ripples ($\mathbf{r_5}$) and edges ($\mathbf{e_5}$):

$$\mathbf{l_5} = (1, 4, 6, 4, 1)^T, \qquad \mathbf{s_5} = (-1, 0, 2, 0, -1)^T,$$
$$\mathbf{r_5} = (1, -4, 6, -4, 1)^T, \qquad \mathbf{e_5} = (-1, -2, 0, 2, 1)^T.$$

The matrix resulting from the multiplication of $\mathbf{l_5}$ with itself is disregarded, because it calculates a weighted average. The final set consists of 15 matrices. The features are defined as the energy of the convolution of the filter matrices with the image. The resulting image $g(u, v)$ is convolved with a Gaussian low-pass filter of size 5×5, referred to as f_1 to decrease high-frequency noise. To combine the $N = 15$ feature images, the average is calculated.

Features in the spatial frequency domain

As the slats can only be placed at certain distances and the tips have a given distance between them, one would expect certain spatial frequencies in the Fourier space to have peaks.

The tips and sinks of a slat form vertical lines in the image and have a certain distance. This results in symmetric horizontal lines with varying intensity in the frequency domain. The distance d_l of the first of the symmetric lines to the axis can be calculated from the vertical distance of two supporting tips d_{tip}, the height of the image h and the size of a pixel Δx [5]: $d_l = \frac{1}{d_{\text{tip}}} \Delta x h$.

These expected horizontal lines can clearly be seen in the frequency domain (see Fig. 3.2 left) and do occur at the predicted

distances. Since there are more lines of higher frequencies, three band-pass filters (BP1 to BP3) are defined to extract the signal in the frequency space. The range of values for f_x is limited by a boundary b_x, as the energy of the signal decreases with higher frequencies.

$$\mathrm{BP1}(f_x, f_y) = \begin{cases} 1, & \text{if } 30 < |f_x| < b_x \text{ and } |f_y| < 5 \\ 0, & \text{otherwise} \end{cases}$$

$$\mathrm{BP2}(f_x, f_y) = \begin{cases} 1, & \text{if } |f_x| < b_x \text{ and} \\ & d_1 - 5 < |f_y| < d_1 + 5 \\ 0, & \text{otherwise} \end{cases}$$

$$\mathrm{BP3}(f_x, f_y) = \begin{cases} 1, & \text{if } |f_x| < b_x \text{ and} \\ & 2 * d_1 - 5 < |f_y| < 2 * d_1 + 5 \\ 0, & \text{otherwise} \end{cases}$$

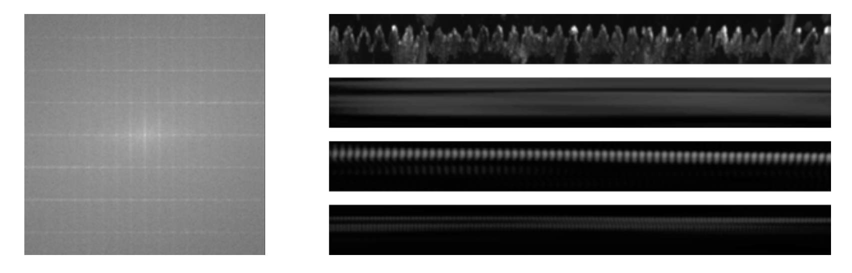


Figure 3.2: Left: An example section of the Fourier magnitude spectrum with the origin in the center. For better visibility of the characteristic features, the leakage effect was reduced by multiplying the original image with a Hann window. Right: An example section of a slat. From top to bottom: pixel values, BP1, BP2 and BP3.

After applying one of the filters, the image is transformed with the inverse DFT, resulting in an image containing only those pixels of the filtered frequency.

3.3 Classifiers

Two classifiers were used to extract the slat positions from the feature images. The parallel projection is intuitive in this case, as the slats are vertical lines in the images. Template matching is a widespread method for object detection and localization.

Parallel Projection

One simple and intuitive way of detecting objects that span across the whole y-direction is parallel projection. Summing the pixel values of the feature image along the y-axis results in a discrete signal that should have peaks at the object locations.

The wavelet transform based pattern matching method presented in [11] is used to extract these peaks. First, a convolution of the signal with discretized Ricker wavelets of 20 different widths is calculated. The response of this convolution is stored in a matrix with each row describing a different wavelet. Next, the peaks in this matrix are analyzed within a row and across adjacent rows. These peaks are referred to as ridge lines. They can be filtered for, giving the position of the peaks.

The advantage of this method is that peaks of different sizes can be detected easily and that the shape information of each peak is not lost, leading to higher information efficiency.

Template Matching

A widely used method for detecting objects in a signal is template matching, which has a two-step approach [12]. Firstly, a template is generated either by example or hand-crafting. Secondly, the occurrence of the template in a given image is evaluated using a similarity measure, e. g. the sum of absolute differences or the normalized cross correlation (NCC) [12]. In this case the NCC was used, since it is almost independent to changes in brightness or contrast of the image [13].

The simplest template to use for this use-case is the image of a new slat. Since the perspective changes from the left to the right of the image, a slat from the middle of the image is taken.

For frequency space features, the template is taken from the filtered and inverse transformed image. A new slat from the middle of the palette is used and will be referred to as “filtered slat template”. A problem occurring quite often with this feature were double peaks, where both the sinks and tips of a slat cause a peak in the NCC. One way to compensate this is to crop the left part of the template since the right side of a correct peak is mostly dark, but to the left there

might be wrongly detected sinks. This template will be referred to as "asymmetric filtered slat template".

The other template used is a simple binary mask, that detects a bright area in the middle of two black areas. The masks must have a suitable width in pixels w_s, given the width of a slat in the image:

$$BM(u,v) = \begin{cases} 1, & \text{if } \frac{1}{3}w_\text{s} < v < \frac{2}{3}w_\text{s} \\ 0, & \text{if } 0 \leq v \leq \frac{1}{3}w_\text{s} \text{ and } \frac{2}{3}w_\text{s} \leq v < w_\text{s} \end{cases}$$

Transformation to slat positions

There is a given number of possible slat positions on the pallet, as slats can only be inserted in their socket. Therefore, a transformation is needed to map from the 3100 image columns to 93 slat socket positions. Since each image might have a slightly different angle or calibration, a general transformation from x-positions to slat positions is not possible. In order to calculate this transformation one needs to extract the position of the sheet stop x_stop from the images. The minimum distance between two slats d is roughly the same as the distance between the slat stop and the first slat. The sheet stop position was extracted using template matching on the original image analogous to the description above in a small area in the upper left region of the image. The template was taken from a single image for each of the two test sites.

The position $n \in \{1, 2, \ldots, 92, 93\}$ of a slat detected at a certain x-position x_n is calculated as:

$$n = \text{round}\left(\frac{x_n - x_\text{stop}}{d}\right) .$$

4 Evaluation

4.1 Results

A test set of 215 images acquired from TRUMPF TruLaser 3000/5000 machines was used to evaluate the methods. 27 images were taken in a TRUMPF test setting and 188 images at a test customer site. The pallets of all machines are almost identical and have 93 slat sockets. Both the slats and the sheet stop were labeled manually.

Table 1: Accuracy results of different features with a parallel projection based classifier.

Feature	True pos. rate	True neg. rate	Accuracy
Pixel values	0.86	0.486	0.667
BP1	0.855	0.412	0.625
BP2	0.916	0.655	0.781
BP3	0.897	0.805	0.85
Laws Energy Measures	0.903	0.453	0.67
Harris Corner Detector	0.605	0.396	0.497
Difference-of-Gaussian	0.88	0.581	0.725
Gradient-of-Gaussian	0.807	0.336	0.562

The evaluation of the methods is based on a binary vector with 93 elements. For every classification result, the accuracy is calculated as:

$$\text{accuracy} = \frac{\text{no. of true pos.} + \text{no. of true neg.}}{\text{no. of classifications}} .$$

As the data set is quite balanced between empty and full slat positions, this metric is applicable.

The true positive rate, true negative rate and accuracy can be seen in Tables 1 and 2.

Table 2: Accuracy of different features with a template matching based classifier.

Feature	Template	True pos. rate	True neg. rate	Accuracy
Pixel values	New slat	0.204	0.642	0.434
BP2	Filtered slat	0.948	0.948	0.948
BP2	Asymmetric filtered slat	**0.966**	**0.958**	**0.961**
BP1	BM	0.764	0.715	0.738
BP2	BM	0.921	0.857	0.887
BP3	BM	0.93	0.87	0.898

4.2 Discussion

Generally, methods using one of the frequency line features performed better than any other method tested. The best method with

a different feature is the DoG with 72.5 % accuracy. Methods using the frequency line features score around 90 % quite often.

Using operations on the pixel values themselves had a better performance with parallel projection than template matching. This seems to indicate that the variation in lighting and wear condition is high. Otherwise it would be easier to find a good template and thus feature images are needed.

Edge and corner detectors as well as Laws' energy measures yield accuracies between 56 % and 72 %. Most likely they are sometimes confused by structures in the background as well as slag formations on the slats. Also, they use no information about tip distances, a definite disadvantage compared to the frequency line features.

Mistakes are mostly made in that part of the pallet, where the slats are seen at an angle, regardless of which method is used. One might expect that this error pattern can be reduced if the template for template matching classifier is taken from this area of the image. A test showed slightly worse results though, most likely because the variations in the appearance of the slats have a greater effect if more of the side of the slat is visible.

5 Conclusion

In this work, different methods for localizing slats of LFMs have been tested on a data set from different machines. Whilst edge and corner detectors, texture measures and operations on the pixel values showed accuracies between 45 % and 70 %, features based on the spatial frequency were best to extract the information and lead to an accuracy of up to 96.1 %.

For this study, neural nets were disregarded, as there are rather few images. With more images from different sites it might become feasible to train neural networks for this task in the hope that they will be better in suppressing relevant noise.

Another direction for future work is the fusion of information across images. In this study, every image was treated by itself. However, since the pallets of a single machine do not change much over time, there might be valuable information that can be extracted from the image sequence.

References

1. F. Struckmeier and F. Puente León, "Nesting in the sheet metal industry: dealing with constraints of flatbed laser-cutting machines," *Procedia Manufacturing*, vol. 85, pp. 149–158, 2019.
2. R. Dewil, P. Vansteenwegen, and D. Cattrysse, "A review of cutting path algorithms for laser cutters," *The International Journal of Advanced Manufacturing Technology*, vol. 87, no. 5-8, pp. 1865–1884, 2016.
3. F. Struckmeier, J. Zhao, and F. Puente León, "Measuring the supporting slats of laser cutting machines using laser triangulation," *The International Journal of Advanced Manufacturing Technology*, vol. 108, no. 11, pp. 3819–3833, 2020.
4. A. Andreopoulos and J. K. Tsotsos, "50 years of object recognition: Directions forward," *Computer Vision and Image Understanding*, vol. 117, no. 8, pp. 827–891, 2013.
5. J. Beyerer, F. Puente León, and C. Frese, *Machine Vision: Automated Visual Inspection: Theory, Practice and Applications*. Berlin, Heidelberg: Springer, 2016.
6. M. Barva, M. Uhercik, J.-M. Mari, J. Kybic, J.-R. Duhamel, H. Liebgott, V. Hlavác, and C. Cachard, "Parallel integral projection transform for straight electrode localization in 3-D ultrasound images," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 55, no. 7, pp. 1559–1569, 2008.
7. D. G. Lowe, "Object recognition from local scale-invariant features." in *IEEE International Conference on Computer Vision*, vol. 99, no. 2, 1999, pp. 1150–1157.
8. D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.
9. C. G. Harris, M. Stephens *et al.*, "A combined corner and edge detector." in *Alvey Vision Conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.
10. K. I. Laws, "Texture energy measures," in *Proceedings: Image Understanding Workshop*, 1979, pp. 47–51.
11. P. Du, W. A. Kibbe, and S. M. Lin, "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching," *Bioinformatics*, vol. 22, no. 17, pp. 2059–2065, 2006.
12. G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11–28, 2016.

13. K. Briechle and U. D. Hanebeck, “Template matching using fast normalized cross correlation,” in *Optical Pattern Recognition XII*, vol. 4387. International Society for Optics and Photonics, 2001, pp. 95–102.

A real-time SAR image processing system for a millimetre wave radar NDT scanner

Christopher Schwäbig, Siying Wang, and Sabine Gütgemann

Fraunhofer-Institute for High Frequency Physics and Radar Techniques
FHR
Department Integrated Circuits and Sensor Systems
Fraunhoferstraße 20, 53343 Wachtberg, Germany

Abstract An ultra high resolution millimetre wave scanner for real-time SAR imaging is being developed at the Fraunhofer Institute for High Frequency Physics and Radar Techniques FHR. Highly integrated radar sensors with ultra wide bandwidth coupled with a new highly efficient SAR signal processing routine incorporate an illuminating scanner system for inline inspection of different materials and goods.

Keywords Extremely high frequency, SAR, real time image processing, CUDA®, NDT

1 Introduction

For various inspection tasks of different goods (for example food, 3D printed plastics) millimetre waves can be used. Millimetre waves depict the range from 30 GHz to 300 GHz of the electromagnetic spectrum. They are able to penetrate plastics, wood, glass and other materials with a low relative permittivity (ε_r). An advantage of millimetre waves in comparison to X-ray (which is also used for many inspection tasks) is the non-ionising radiation meaning that no special protection is necessary during the operation.

2 Current development status

The standalone millimetre wave imager (“SAMMI”) [1] works with two antennas for transmitting and two antennas for receiving the

electromagnetic wave. The transmitting antennas are placed underneath and the receiving antennas above a conveyor belt (see figure 2.1).

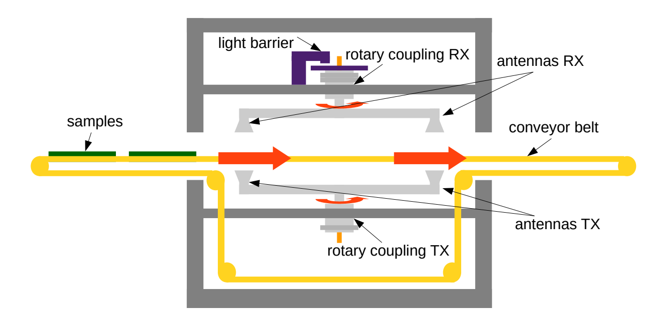


Figure 2.1: Mechanical concept of the current SAMMI version

A continuous wave signal at a frequency of 90 GHz is generated by a chain of several frequency mixers, filters and amplifiers. While the antenna pairs are moving along the conveyor belt (detected by a light barrier) an analogue digital converter samples the referenced raw data of the received signal and transfers it to the computer, where the transmission images are calculated and dynamically displayed in the user interface. These images are a 2D amplitude and a 2D phase image which represent the attenuation and the runtime of the electromagnetic wave within the object. SAMMI demonstrates the efficiency of radar technology and opens up development potential for further applications. To achieve a higher level of integration and above all, the possibility of the 3D imaging for nondestructive testing (NDT), an advanced system is developed as described in the following.

3 Development of a new SAMMI

As a result of these considerations a new system with a higher integration level, less mechanical expense and a frequency modulated radar is being developed (see figure 3.1). This enables the system to additionally provide depth information for real 3D imaging.

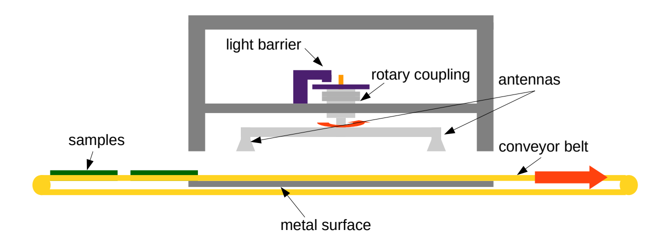


Figure 3.1: Mechanical concept of the SAR-SAMMI

The concept of this new SAMMI generation is based on synthetic-aperture radar (SAR) which is a retro-reflexive measurement method. For this approach two antennas above the conveyor belt are sufficient. In consequence the whole mechanical setup can be reduced so that mass and dimensions of the demonstrator decrease. Artefacts in the image due to the less ideal synchronicity of the antenna pairs do not exist. In contrast to the previous system the image processing algorithm becomes more complex.

4 Hardware signal processing

Radar sensors can be used for various applications for non-destructive testing [2], e. g. thickness measurement of multilayer samples [3] or determination of electromagnetic material properties. A FMCW radar based on a highly integrated SiGe radar chip [4] is used. The measurement principle is an indirect time-of-flight estimation of the difference between the transmitted and the reflected electromagnetic waves. For short range application the very high bandwidth of the radar chip allows ultra high resolution SAR imaging. During the measurement the received signals are collected, digitised and transferred to the computer (see figure 4.1).

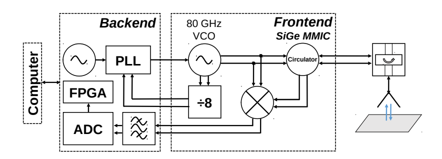


Figure 4.1: Hardware block diagram of the SAR-SAMMI

5 Image reconstruction signal processing chain and mathematical background

A highly efficient SAR algorithm is developed and implemented. During one single semi circular movement of the antenna 192 FMCW sweeps are recorded. The calculated results obtained from the data of one semi circular movement are stored within a temporary 3D array which is inserted in the final 3D output volume after the data of all sweeps have been processed.

5.1 Precalculations (only made once before the measurement)

The antenna positions during the circular antenna movement are presumed as constant so that a precalculation of all distances between the antenna positions and the voxels of the temporary 3D array can be done. Therefore calculation time for a real-time implementation decreases.

A precalculated mask for each sweep of the semi circle reveals whether or not the voxel of the temporary 3D volume can receive information of the current sweep. A look up table which is precalculated for each sweep of the semi circle, contains the distance for each voxel which has to be analysed (green in figure 5.1) in the sweeps spectrum.

5.2 Analysing the sweep of the semi circle

Each single sweep of the semi circle is filtered with a Hamming window in order to reduce the spectral leakage. FFT interpolation is used to achieve a higher resolution of the sweep signal in the frequency domain. If a voxel of the temporary 3D volume can receive

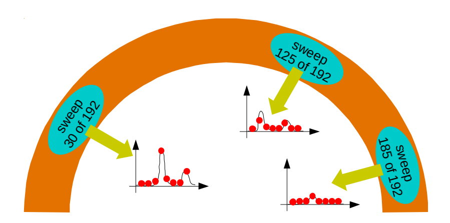


Figure 5.1: The sphere of influence (green) of different sweeps within the semi circle (orange)

information of the current sweep, the look up table and the nearest neighbour method are used to extract the image information from the sweeps spectrum. Based on the conventional backprojection algorithm [5] these values are weighted with a complex exponential function (see formula 5.1) dependent on their distance and are added into the temporary 3D volume (see figure 5.2).

$$s(m, \tau_n) = \texttt{fft}(s(f_k, \tau_n)) \cdot \exp \frac{+j4\pi f_1 \Delta R(m, \tau_n)}{c_0} \tag{5.1}$$

$s(m, \tau_n)$ describes the weighted value of *one* sweep for one voxel within the temporary 3D volume, dependent on the quantised range bin m within the sweeps spectrum and the time τ_n which depends on the number of the sweep. $s(f_k, \tau_n)$ describes the received quantised signal for one single sweep with k frequencies (f). $\Delta R(m, \tau_n)$ describes the quantised distance to the appropriated voxel. The frequency f_1 represents the start frequency of the FMCW down sweep or FMCW up sweep. For calculating the final value of one voxel of the temporary 3D volume, the values for $s(m, \tau_n)$ of all 192 sweeps have to be summed up.

5.3 Stepwise image build-up

After all sweeps of the semi circle have been processed, this temporary 3D volume is added into the final 3D volume. As a result of the movement of the conveyor belt the final 3D volume is shifted before the values of the temporary 3D volume are added. In order to create a x, y and z perspective of the final 3D volume, the amplitude values

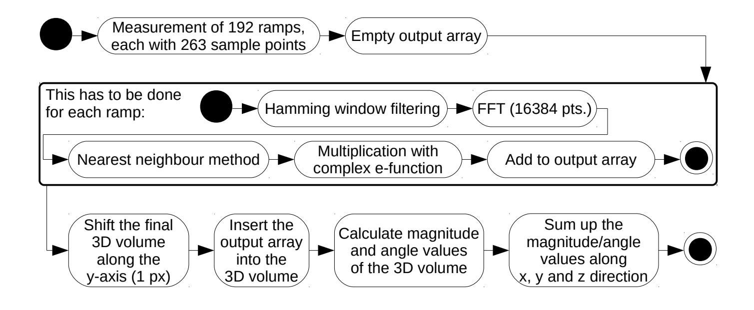


Figure 5.2: Software signal processing chain for each semi circle

are summed up along each axis. A whole image is created, based on multiple semi circles and the movement of the object (see figure 7.2). The speed of the conveyor belt depends on the chosen voxel size because the rotation speed of the antenna is assumed to be fixed.

6 Implementation

The implementation is divided into a CPU and a GPU part. The CPU part is separated into three threads and is responsible for GUI/visualisation, data capture (with hardware communication) and CPU-GPU data transfer. On the GPU the whole image processing chain is processed so that the array operations are calculated in parallel. Therefore the array operations are handled element by element and each array position is processed by its own thread within the GPU.

Before the actual measurement data evaluation can be processed, the precalculated look up tables are copied to the GPU. For each sweep (192 in total) of the semi circle two look up tables are computed. The first look up table illustrates whether a voxel within the temporary 3D volume can receive information from the current sweep or not. If a voxel can receive information from the current sweep the second look up table is used. This second look up table shows the distances (converted into FFT positions) for every voxel of the temporary 3D volume, which have to be analysed within the spectrum.

After these look up tables have been copied to the GPU the actual measurement process can be started. In this case a memory transfer (of precalculated data) is no longer necessary during the measurement which improves the execution time of the whole signal processing chain.

The different steps of the signal processing chain (see figure 5.2) are implemented in single functions (kernels). The batched CUDA® FFT routine (and respectively in general the CUDA® FFT) is called up by the host (the computer) and not by the device (the graphics card) itself. In connection with this, the other parts of the signal processing chain are implemented in the same way so that different CUDA® kernels represent the single steps of the signal processing chain. The sizes of the arrays (which are processed within the different functions) vary in size. In relation to this most of the different kernels (functions) vary in their thread and block configuration.

6.1 Detailed description of the single kernels within the signal processing chain

In the initial step the raw data of all 192 sweeps is copied to the GPU with help of the CUDA® kernel "cudaMemcpyAsync". Afterwards the raw data of all 192 sweeps is resorted from the hardware arrangement to a new arrangement so that the data can be directly processed with the batched FFT ("prepareDataForBatchedFFTWithWindowFilter"). In addition to this the raw data is filtered with a Hamming window. After the batched 1D FFT has been executed the temporary 3D volume is emptied (CUDA® kernel "cudaMemsetAsync"). This is necessary because this temporary 3D volume contains values of the previous semi circle. In the next steps this temporary 3D volume will be filled with processed data of the current semi circle.

The look up table and the nearest neighbour method are used to extract the necessary information from the corresponding spectrum ("nearestNeighbourMethodWithSum"). The extracted values of all 192 sweeps are summed up within the temporary 3D volume. Before the temporary 3D volume is inserted, the kernel "copyValues" creates a copy of the original final output 3D volume and the kernel "shiftValues" writes the values shifted by one element along the y axis in the original final output 3D volume. After the temporary 3D

volume has been inserted (“insertMeasurement”) the absolute values of the complex final output 3D volume are calculated (“complex-ToAbs”). Based on the final output 3D volume with absolute values, three 2D images (one with the sums along the x-, one with the sums along the y-, and one with the sums along the z-axis) are calculated (“calculateXSum” / “calculateYSum” / “calculateZSum”). Within the last step these three 2D images are copied to the CPU (CUDA® kernel “cudaMemcpyAsync”).

6.2 Memory usage

Shared memory is useful when data stored within the global memory has to be accessed multiple times or data elements have to be shared between different threads in one block. In this implementation the implemented kernels are only responsible for a simple array operation and multiple access is not necessary. Therefore the use of shared memory by copying the data from the global memory into the shared memory is waived.

By using coalesced access to the global memory the execution time is much faster compared to uncoalesced access. In this case most of the kernels use a coalesced access to the memory in order to improve the speed of the algorithm. The kernels “prepare-DataForBatchedFFTWithWindowFilter” and “nearestNeighbourOptAllRampsWithSum” are much slower than the other kernels because their memory access is uncoalesced. The reason for this is that the kernel “prepareDataForBatchedFFTWithWindowFilter” resorts the input data (the algorithm has no bearing on the arrangement of this data) and the kernel “nearestNeighbourOptAllRampsWithSum” has to access various elements within the FFT spectrum (uncoalesced reading, but coalesced writing).

The implementation makes use of pinned memory in order to optimise the execution speed of the algorithm. In this case the recorded raw data is stored directly within the pinned memory. The same applies to the data which contain the three output images of the algorithm. In total, four arrays are declared as pinned memory (raw data input, output image x, output image y and output image z direction).

6.3 Further implementation details

In order to calculate the multiple 1D FFTs (192 in total) of one semi circle at once and not 192 single 1D FFTs this implementation makes use of the CUDA® batched 1D FFT. The advantage is that the execution routine of the FFT plan has only to be called once and not 192 times. The FFT plan consists the FFT settings and is created before the actual measurement is processed.

For the purpose of allowing further signalprocessing steps (for example phase unwrapping) the implementation takes advantage of CUDA® streams so that the semi circles can be processed in parallel by putting them into different GPU streams. To achieve this each signalprocessing chain for one stream has to be controlled by its own CPU thread. In this case each GPU stream is mapped with its separate plan of a batched 1D FFT.

The asynchronous memory transfer function "cudaMemcpy-Async" (host to device or device to host) is used so that each CPU thread can copy the data within its corresponding GPU stream. The same applies to the emptying process of the temporary 3D volume (before the steps of the signal processing are executed) with the function "cudaMemsetAsync".

Ideally the GPU streams are all processed in parallel. If the computing capacity of the operating graphics card is insufficient, parts of the processing run in sequence and in consequence the runtime increases.

7 Results

The created amplitude image (see figure 7.2, left) is based on simulated raw data of a 100 mm $\times$ 100 mm metal Siemens star (see figure 7.1).

In the simulated raw data the object is placed at a height of 30 mm which can be seen in both side views. The simulated raw data consists of 80 lines with 192 FMCW sweeps each (low resolution mode) and 160 lines with 192 FMCW sweeps each (high resolution mode). The raw data takes into account the semi circular movement path of the antenna. With the help of the algorithm this circular movement is corrected so that the measured object is depicted in the

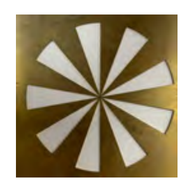


Figure 7.1: Photo of a Siemens star

correct shape and aspect ratio. More details concerning the object can be acquired by analysing the phase values which requires a 2D phase unwrapping algorithm [6].

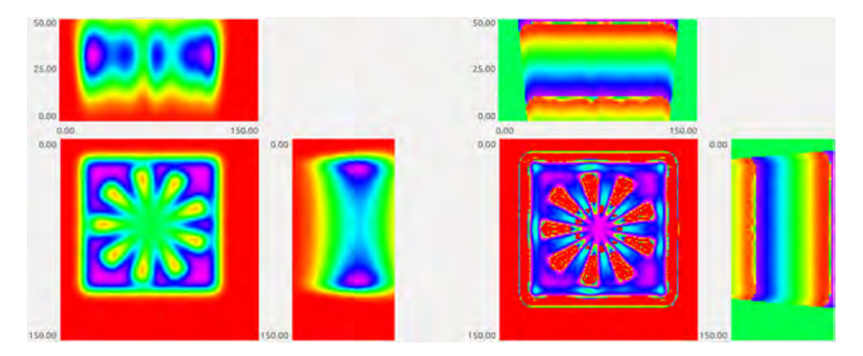


Figure 7.2: x, y and z view of the amplitude (left) and phase (right) 3D volume of a Siemens star

Because the Siemens star simulated here is infinitely thin, a top view (z axis) of the Siemens star can be generated without using a 2D phase unwrapping algorithm (see figure 7.2, right). The side views of the 3D phase image shows that the phase values pass through the whole phase spectrum.

7.1 Kernel runtimes

Table 1 shows the execution times of the different kernels of the signal processing chain for a low and high resolution compared on two graphics cards. It is obvious that the data preparation for the FFT,

the FFT itself and the nearest neighbour method (with sum) consume most of the time (marked in red).

Table 1: Benchmark

Function / kernel	NVIDIA® Quadro™P400 (2GB)		NVIDIA® Quadro RTX™6000 (24GB)	
	Low resolution	*High resolution*	*Low resolution*	*High resolution*
Memcpy HtoD (async)	85.983 µs	n.p.[a]	45.056 µs	44.939 µs
prepareDataForBatchedFFT..	932.689 µs	n.p.[a]	38.496 µs	38.592 µs
Batched FFT	4183 µs	n.p.[a]	204.801 µs	202.113 µs
memset (Empty array)	24.256 µs	n.p.[a]	2.528 µs	21.28 µs
nearestNeighbourMethodWithSum	5340 µs	n.p.[a]	214.242 µs	2650.87 µs
copyValues	134.174 µs	n.p.[a]	5.696 µs	98.785 µs
shiftValues	398.362 µs	n.p.[a]	15.872 µs	273.826 µs
insertMeasurement	367.418 µs	n.p.[a]	9.28 µs	133.772 µs
complexToAbs	398.714 µs	n.p.[a]	13.408 µs	218.274 µs
calculateXSum	132.446 µs	n.p.[a]	17.568 µs	219.905 µs
Memcpy DtoH (async)	2.432 µs	n.p.[a]	1.472 µs	5.216 µs
calculateYSum	294.396 µs	n.p.[a]	29.568 µs	250.114 µs
Memcpy DtoH (async)	2.848 µs	n.p.[a]	1.44 µs	5.248 µs
calculateZSum	177.597 µs	n.p.[a]	10.272 µs	241.378 µs
Memcpy DtoH (async)	10.592 µs	n.p.[a]	5.216 µs	29.344 µs

[a] Not possible.

7.2 Memory and GPU utilisation

In the high resolution mode the required memory of the three dimensional arrays is around 15 times larger compared to the low resolution mode. The final three dimensional volume in the low resolution mode consists of 248897 values compared to 3624040 values in high resolution mode. The temporary three dimensional volume of one sweep in the low resolution mode consists of 100793 values compared to 1444800 values in high resolution mode.

In table 2 the memory usage and GPU utilisation during the measurement is shown. For the low resolution mode (2.5 mm), approx. 0.7 GB is used for the memory of the precalculated arrays and all the intermediate steps of the signal processing chain. Therefore on a small graphics card (for example NVIDIA® Quadro™P400, 2GB) 35 % of the memory is used. By using this graphics card the high resolution mode is not executable because of the higher memory requirements (10 GB).

The GPU utilisation is measured for four different input data rates. In the current implementation the use of CUDA® streams is not necessary because the input data rate of semi circles (10 Hz) is a lot

lower than the maximum possible frame rate. Therefore the number of streams can be reduced to one.

Table 2: Memory usage and GPU utilisation

	NVIDIA® Quadro™P400 (2GB)		NVIDIA® Quadro RTX™6000 (24GB)	
	Low resolution	*High resolution*	*Low resolution*	*High resolution*
Memory signal processing	0.7 GB (35 %)	10 GB (500 %, n.p.[a])	0.7 GB (3 %)	10 GB (42 %)
Memory operating system	430 MB (22 %)	430 MB (22 %)	430 MB (2 %)	430 MB (2 %)
GPU utilisation	15 % (10 Hz) 30 % (20 Hz) 55 % (40 Hz) 90 % (67 Hz, max.)	n.p.[a] n.p.[a] n.p.[a] n.p.[a]	2 % (10 Hz) 3 % (20 Hz) 6 % (40 Hz) 60 % (350 Hz, max.)	6 % (10 Hz) 11 % (20 Hz) 22 % (40 Hz) 65 % (125 Hz, max.)

[a] Not possible.

Table 3 shows the execution speed of the different implementations for the low resolution and high resolution mode.

Table 3: Speed comparison

	Low resolution	*High resolution*
GNU Octave, CPU, one core	1.43 Hz	0.14 Hz
C Implement., CPU, one core	12 Hz	1.25 Hz
NVIDIA® Quadro™P400	67 Hz	n.p.[a]
NVIDIA® Quadro RTX™6000	350 Hz	125 Hz

[a] Not possible.

8 Conclusion

With the help of SAR it is possible to reduce the mechanical setup of the original standalone millimetre wave imager. By implementing the signal processing in CUDA® a real-time image generation is possible. Due to the fact that the achieved throughput is much higher than necessary, there is sufficient capacity for further signal processing steps (for example phase unwrapping).

References

1. A. Küter, C. Schwäbig, R. Brauns, S. Kose, and D. Nüßler, "A stand alone millimetre wave imaging scanner: System design and image anal-

ysis setup," *48th European Microwave Conference (EuMC), pp. 1505–1508*, 2018.

2. R. Herschel and S. Pawliczek, "3D millimeter wave screening of wind turbine blade segments," *15th European Radar Conference (EuRAD), pp. 115–117*, 2018.
3. S. Pawliczek, R. Herschel, and N. Pohl, "High precision surface reconstruction based on coherent near field synthetic aperture radar scans," *19th International Radar Symposium (IRS)*, 2018.
4. C. Bredendiek, K. Aufinger, and N. Pohl, "Full waveguide E- and W-band fundamental VCOs in SiGe:C technology for next generation FMCW radars sensors," *14th European Microwave Integrated Circuits Conference (EuMIC)*, 2019.
5. L. A. Gorham and L. J. Moore, "SAR image formation toolbox for MATLAB," *Algorithms for Synthetic Aperture Radar Imagery XVII*, 2010.
6. S. Pawliczek, R. Herschel, and N. Pohl, "3D millimeter wave screening for metallic surface defect detection," *16th European Radar Conference (EuRAD), pp. 113–116*, 2019.

Towards a remote EEG for use in robotic sensors

Niels-Ole Rohweder[1], Christian Rembe[1], and Jan Gertheiss[2]

[1] Clausthal University of Technology,
Institute for Electrical Information Technology,
Leibnizstrasse 28, 38678 Clausthal-Zellerfeld
[2] Helmut Schmidt University, Department of Mathematics and Statistics,
Holstenhofweg 85, 22008 Hamburg

Abstract Recently, it was shown that a correlation exists between brain activity and oscillations of the pupil. As the experiment was designed to measure exitations of the pupil for frequencies below 1 Hz, whether such correlations also exist on the scales of seconds and for frequencies between 5 and 40 Hz is still an open question. In this work, we design a new experiment and measure the response of the pupil to continuous, periodic visual and acoustic stimuli. We show that a clear response of the pupil for flashes of 7.5 Hz exists, bearing similarity to the effect known as Steady-State Visual Evoked Potential in neuroscience. This result can directly be used to develop a new kind of non-contact brain-computer-interface, using visual fixation as a trigger. Further, we evaluate the pupil response to series of white noise clicks with a frequency of 8 Hz, in order to assert the pupil response as due to brain activity. First results indicate the presence of a weak signal, showing the stimulus frequency and harmonics, bearing similarity to the neural effect known as Auditory Steady-State Response. Measuring brain activity remotely could provide pathways to new kinds of sensors, in particular for collaborative robots and general human-machine-teams, where estimates of the mental state of the human partner are essential.

Keywords eeg, remote eeg, pupil, oscillations, SSVEP, ASSR, HMT, BCI, sensor

1 Introduction

One central aspect of Human-Machine-Teams (HMT) – be it collaborative work in factories, or driving partly autonomous cars – is the ability of the machine to evaluate its human partner. Especially in safety-critical environments, a precise state model of the human is essential, consisting, at a minimum, of a binary attribute of "take-over-readiness" (TOR) – the ability of the human to perform the required task, e. g. taking back control of the steering wheel, or accepting a hazardous object in a factory. Otherwise, the machine or robot is left blind; and indeed, that is the current state-of-the-art. In collaborative situations such as autonomous cars, the focus is placed on clear interfaces to signal the human partner the need to take over, without making sure they are actually able to do so [1]. Even the very term of "take-over-readiness" and the concept it describes is used solely in the context of partly autonomous cars, not found anywhere else, and the application of human models and considerations of the HMT as one unit, i. e. a human-in-the-loop approach, are only of very recent focus in the literature [2,3].

Regardless of the complexity of the human model – a single attribute or a full Theory of Mind – the required sensor input is going to consist of both physical and mental parameters. While research and sensors for both exist, only physical parameters (e. g. hand position, heart rate) so far have been measured remotely. The gold standard for measuring mental parameters such as situation awareness, cognitive load or tiredness, the electroencephalogram (EEG), requires electrodes placed on the scalp for good signal quality. Obviously, such a setup is infeasible in real world applications, outside of very special circumstances. On the other hand, Park and Whang recently showed that a correlation exists between brain activity and pupil size, such that the electrical state changes created by the neurons – and measured as oscillations of electrical potential on the skin – are mirrored by oscillations of the size of the pupil [4]. In particular, they showed a strong correlation of activity in the front and central brain region with pupil oscillations, e. g. in the mu-band around 10 Hz, and the gamma-band between 30 and 50 Hz.

However, the setup of Park and Whang used long-time averages, comparing the frequency bands of the EEG with 1/100 subharmon-

ics for the pupil oscillations (e. g. the 10-Hz-band of the former with pupil oscillations around 0.1 Hz), thus creating correlations on the scales of minutes. A natural extension of the experiment is to ask whether such correlations also can be measured directly, using the same frequency bands, thus increasing the time resolution to seconds, rather than minutes. A second question is whether such correlations, if they exist, can be used to create a reliable remote EEG, for use in sensors to serve as input of a human model, e. g. in the context of deriving a measure of “trust” between the partners [5]. This overachieving question is the purpose of the HerMes project of the Clausthal University of Technology, in the context of which our experiments are performed [6].

2 Steady-State Evoked Potentials as stimuli in experiments

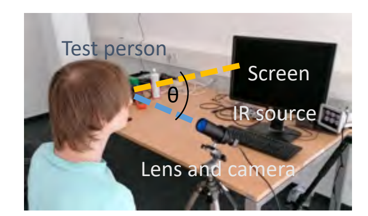
From the outset, it is clear that measuring oscillations of frequencies higher than $\sim$1 Hz decreases the resulting amplitude drastically. The pupil is an imperfect oscillator, the speed of the dilation or contraction which the iris muscles can achieve is limited, hence the response to any stimuli is limited as well. Literature confirms this hypothesis [7,8]. In order to increase the signal-to-noise ratio, it is therefore desirable to have an artificially triggered, continuous signal that can clearly be separated from the noise and measured for arbitrary durations. In neuroscience, such signals are known as “Steady-State Evoked Potentials”, a response of the brain to a continuous sensory stimulus at a certain frequency. The stimulus frequency, typically including harmonics, can be clearly measured in the brain activity, for as long as the stimulus persists [9].

The visual version – the Steady-State Visual Evoked Potential, SSVEP – is the one most easily measured. It has an excellent signal-to-noise ratio (SNR) [10]. It is easily triggered e. g. by flashing LEDs or flickering computer screens, and often used for brain-computer-interfaces [11]. The acoustic variant – Auditory Steady-State Response, ASSR – has at least an order of magnitude smaller responses compared to the SSVEP, and is comparatively harder to measure, typically averaged over multiple traces or long periods to create a

sufficient SNR. Generally, the strongest responses appear at and below frequencies of 40 Hz [12]. Both visual and acoustic stimuli were identified as candidates to measure a response in the pupil oscillations.

3 Algorithm

In order to resolve such small pupil oscillations, a precise computer algorithm for evaluating the pupil diameter is needed. Typically, a circle detection is used. Yet unless the camera is placed such that it is perpendicular to the pupil, this is an approximation; the perspective distortion turns the circle into an ellipse of increasing eccentricity for increasing angles. In experiments such as Park and Whang's, as well as in commercial eye-trackers such as Tobii [13], this "pupil foreshortening error" is usually ignored; or avoided, by using large distances between eye and camera. Protocols for post-hoc correction exist [14]. On the other hand, the most convenient placement of the camera for high resolutions of the pupil is close to it, and out of the line of sight, i. e. nearly always at an angle, e. g. looking up from below (Fig. 3.1).



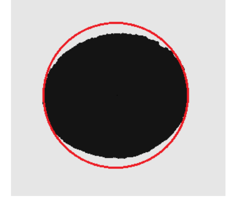



Figure 3.1: Left: The setup of our experiment. Centre: After thresholding to binarise the image of the eye, major and minor axes of the black pupil are detected. The aspect ratio in this image is 1.17, corresponding to an angle θ of approx. 31° for the camera inclination. Right: The perspective distortion causes the pupil to deviate from the ideal circle.

Thus the experiment was planned to improve the previous setups, using elliptical pupil tracking from the start. However, while increasing the accuracy, it also increases complexity. A circle detection algorithm, e. g. the commonly used Hough-transform, operates in a

three-dimensional parameter space, corresponding to the three parameters defining the circle: The x and y component of the centre, as well as the radius. Conversely, an ellipse creates a five-dimensional parameter space, introducing two radii (or axes) and a rotation angle, in addition to the centre coordinates. In order to calculate the relevant quantity, the long (semi-)axis, we use an approach derived from [15, 16]. From the picture to the result, the algorithm works as follows.

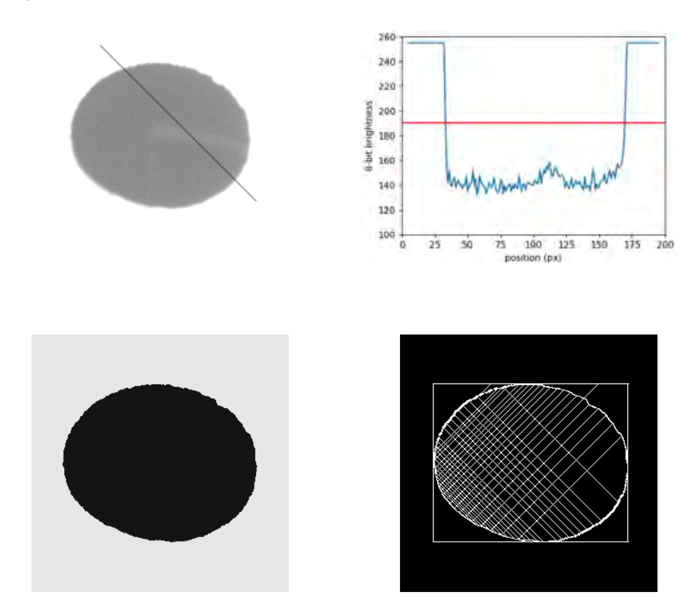


Figure 3.2: Steps of the algorithm. Top left: Image of pupil. A reflection of the experiment's screen is faintly visible, the line indicates a cross-section. Top right: Cross-section along the line with binarising threshold in red. Bottom left: Binarised image. Bottom right: Final result of the Canny edge detection, with the chords used to calculate the ellipse centre shown for illustrative purposes. Chords exceeding the box of the ellipse such as near the upper right corner, due to e. g. missing edge pixels, are discarded.

As the setup is such that the greyscale-image of eye results in the pupil being darker than the rest of the image (see Sec. 4), it is rather easy to binarise, leaving only the pupil behind (Fig. 3.2). Over the binarised image runs a conventional Canny algorithm, which filters

out the edge pixels. The interior is then divided by two sets of chords, running from one side of the ellipse to the other, starting and ending at the innermost pixels. The number of the chords scales with the size of the ellipse, at a typical size of around 20 for each set. Start and end points of the chords create a set M of edge points, defining the ellipse. In accordance with [15,16], we use a geometrical property to determine the centre of the ellipse: The bisections of two sets of chords intersect at the centre of the ellipse. Using a linear regression over the individual centre pixels of a set of chords provides an accurate estimate of its bisector, and leads to a very good estimate of the ellipse centre.

After determining the centre coordinates, the remaining parameter space is three-dimensional once more, and the other parameters can be solved iteratively and algebraically, using three points of the set M. To that end, M is sorted by quadrants and the three points taken from three different quadrants. These three points need to be far from respective points of symmetry that would result in an indetermined ellipse, such as would be the case e. g. with two points $(X|Y), (-X|-Y)$ as measured from the ellipse centre. The average of the result from each of the sets of three is finally taken to get the desired result, the value of the long and the short axis.

The aim here is real-time capability; currently, the entire algorithm, from saving the image to getting the axes lengths, runs at approximately 20 fps on an Intel core i5 (8th gen) processor.

4 Experimental setup

The three parts of the experiment are the test person, the camera and the source of light. The test person takes a seat in front the computer screen. The eye is filmed from below, while the look fixes ahead, on a mark on the screen. The illumination comes from the side (Fig. 3.1).

There are two ways to create a high contrast between pupil and the rest of the eye or face. For the bright pupil effect, in which light is reflected back from the retina into the camera (the same as the "red-eye effect" in photography with a flash), camera and source of light are required to be closely aligned on one optical axis. This in turn

requires a sufficient distance of the eye from the camera, typically on the order of metres, as the dimensions of the camera and the source of light limit their proximity. The dark pupil effect does the inverse, placing the source of light off-angle; the light is reflected away from the camera, and the pupil appears black. This is the only feasible option, as the camera is placed less than half a metre away from the eye in order to create a sufficient resolution of the pupil.

As usual to avoid glare and to provide uniform illumination, near infrared light is used. A simple 6 W LED floodlight of 850 nm wavelength is placed such that the specular reflection off the cornea is not directed towards the camera, avoiding a bright glint in the otherwise dark pupil. The lens in front of the camera is assembled using two Near-IR coated lenses of $f = 150$ and $f = 25.4$ mm focal length and an iris diaphragm in their common focal point, creating a simple telecentric lens. The advantage of such a setup is the independence of the magnification from the distance of the object.

On the one hand, this avoids changes in pupil size by involuntary movements of the head, and on the other hand allows to calibrate the optical system such that the pupil oscillations can be examined in real units, not pixels. As the amplitude of such oscillations has never been measured and therefore is unknown, this was an important consideration. The calibration was performed before the experiment using an USAF target, and determined as 20.83 ± 0.12 µm per pixel. The lens is mounted on a camera with a 3.2 MP resolution (IDS UI-3270CP), of which a field of view of 1024 by 1024 pixels is used, allowing for frame rates of 83 fps. Consequently, frequencies up to 40 Hz can be resolved.

5 Results and discussion

For the first part of the experiment, an SSVEP sequence is displayed on the screen. It consists of 16 seconds of black screen, then 10 seconds of a white flashing box, and an additional 10 seconds of black screen at the end. The frequencies of the flashes are chosen such that the refresh rate of the screen can be synced, i. e. 30 Hz, 15 Hz, 10 Hz or 7.5 Hz. An exemplary result is displayed in Fig. 5.1 for a stimulation frequency of 7.5 Hz. Both the SSVEP stretch and the

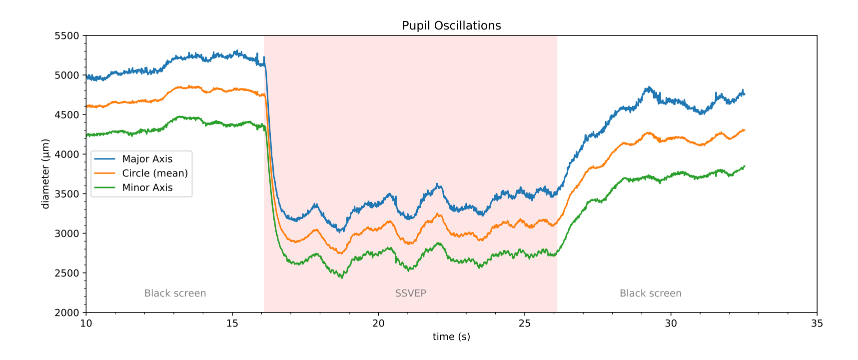


Figure 5.1: Pupil diameter before, during, and after a sequence. The SSVEP stretch is highlighted.

dark screen can be clearly determined by the pupil size. The relative brighter stretch of the flickering screen causes the pupil to contract. A Fourier analysis (resolution bandwidth: 0.125 Hz) of the SSVEP-stretch yields a clear peak at the stimulation frequency of 7.5 Hz, as well as one at its third harmonic, 22.5 Hz (Fig. 5.2, left). The signal-to-noise ratio of the fundamental was 17 dB. The amplitude of the oscillations is on the order of 10 μm. As one pixel was calibrated to 20.83 μm, this is a sub-pixel resolution, a result of the parameter estimation algorithm and the Fourier Transform over a sufficiently long time interval.

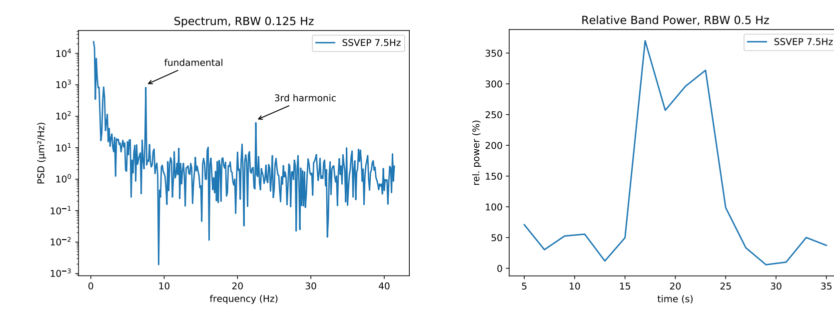


Figure 5.2: Left: Power spectral density of the recorded SSVEP sequence with a frequency of 7.5 Hz. Right: Relative power of the 7.5 Hz band over time, with respect to the mean power of four bands (7.5 Hz, 10 Hz, 15 Hz, 30 Hz).

By calculating the mean power of the four bands (30 Hz, 15 Hz, 10 Hz and 7.5 Hz, resolution bandwidth 0.5 Hz), and looking at the relative power of the respective individual bands with regards to that mean, Fourier transforming intervals as short as 2 seconds resulted in observable pupil responses (Fig. 5.2, right). This result has direct relevance for creating new kinds of non-contact brain-computer-interfaces. By focusing on one of multiple flickering spots, each with a different frequency, and using a threshold value on such a time series of band power, the resulting potential can trigger actions, e. g. controlling disability aids such as wheelchairs [11]. The idea is similar to [7], who suggested such a scheme, at lower frequencies, for tracking attention or focus.

Unfortunately, for visible stimuli, it is hard to separate oscillation induced via the steady-state potential and brain activity from oscillations due to the simple pupillary light reflex. The amplitude of either oscillation is limited by the mechanical constraints of the iris muscle at any rate; biological considerations such frequency-dependent light reflex responses and latencies to estimate the influence on the measurements would appear to increase the complexity of the experiment severely. Instead, we chose to investigate the response of the pupil to acoustic stimuli. This creates a clear distinction between light reflex and brain activity-induced oscillations; however, as noted earlier, the ASSR effect is at least an order of magnitude smaller than its optical counterpart, and therefore harder to isolate from the noise.

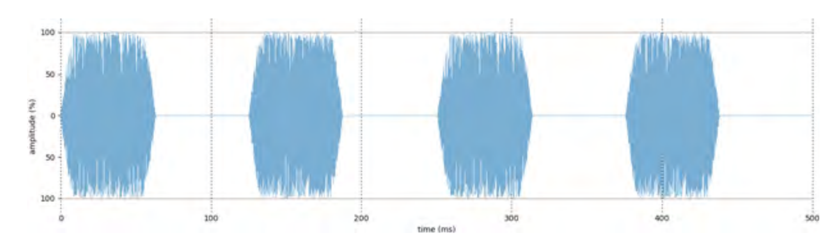
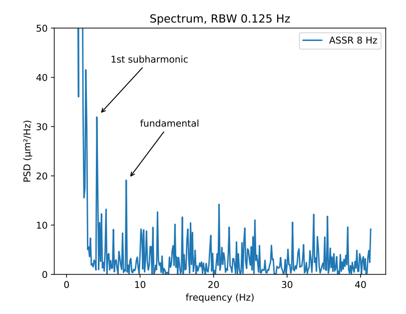


Figure 5.3: The ASSR stimulus. White noise, amplitude-modulated with a rectangle wave of 8 Hz. The modulation depth is 100%.

We used series of white noise click trains, that is, a 100% amplitude modulated sequence of white noise lasting ten seconds and incorporating 80 clicks, creating a rectangle wave of 8 Hz (Fig. 5.3).

The soundfile is played using headphones, while the look of the test person is fixed ahead onto a permanent mark on the screen. The rest of the procedure, as well as the evaluation of the recorded images are exactly as above, using Fourier transforms with a resolution bandwidth of 0.125 Hz.



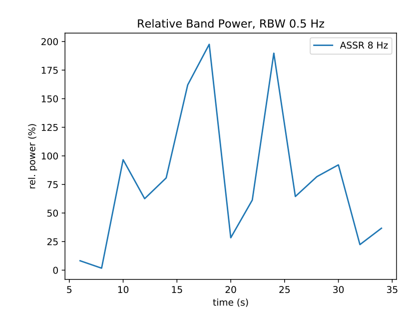


Figure 5.4: Left: Power spectral density of the recorded ASSR sequence with a frequency of 8 Hz. Right: Relative power of the 8 Hz band over time, with respect to the mean power of four bands (8 Hz, 10 Hz, 15 Hz, 30 Hz).

First results indicate the presence of the AM stimulus frequencies and (sub-)harmonics (Fig. 5.4, left); however, the recordings are subject to a lot of noise, often entirely covering the signal. The time series of the relative band power (Fig. 5.4, right) indicates the this: The sequence starts after 15 seconds, and for the 10 second duration, the signal is sporadic, fading entirely before reappearing. The signal-to-noise ratio never exceeds 3 dB, the measured amplitudes of those oscillations are below 4 µm. Interestingly, for highly eccentric ellipses, the oscillations, if they do appear, are visible in the diameter of the long axis only, not in the short axis, likely because the squeezed amplitude due to the perspective distortion is too small to resolve. This justifies the use of the elliptical fit.

Nevertheless, further improvements to both the algorithm and the setup may be needed, to increase resolution and precision and decrease noise, and achieve robust results. A commercial EEG to measure brain activity directly, which so far has not been used, will create another way to confirm the correlation.

6 Summary

A remote EEG is a promising way to create a sensor for use in Human-Machine-Teams. Building on the work of Park and Whang, we developed an experiment to show a correlation between brain activity and oscillations of the pupil diameter. As opposed to Park and Whang, we tried measuring the oscillations in the same frequency band as the brain activity, not its subharmonics, as well as in real units, not pixels. In order to achieve the required precision, we placed the camera in close proximity to the eye for a high resolution of the pupil, and developed an elliptical fitting algorithm in order to compensate perspective distortion. Using both visual and acoustic stimuli – SSVEP and ASSR – we tried to measure the corresponding excitation of the pupil. For the visual stimulation, we received a clear and reliable signal; oscillations of the pupil of approximately 10 µm for a stimulation frequency of 7.5 Hz with a SNR of 17 dB. Using intervals for the Fourier transform as short as 2 seconds, we created a time series of the band power, showing the onset as well as the end of the stimulation, thus allowing for the construction of non-contact brain-computer interfaces. In order to separate the brain activity-induced oscillations from the pupillary light reflex, we also used acoustic stimuli. First results indicate a positive response, showing the stimulation frequency of 8 Hz as well as their (sub-)harmonics; however, subject to much noise, sometimes blanketing the signal, and never exceeding an SNR of 3 dB. The corresponding amplitude of the oscillation is below 4 µm. In the future, decreasing the noise floor and correlating the pupil spectra with commercial EEG spectra could yield more robust results.

References

1. C. Braunagel, W. Rosenstiel, and E. Kasneci, “Ready for take-over? A new driver assistance system for an automated classification of driver take-over readiness,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 4, pp. 10–22, 2017.
2. N. Deo and M. M. Trivedi, “Looking at the Driver/Rider in Autonomous Vehicles to Predict Take-Over Readiness,” *IEEE Transactions on Intelligent*

Vehicles, vol. 5, no. 1, pp. 41–52, 2019, conference Name: IEEE Transactions on Intelligent Vehicles.

3. T. Mioch, L. Kroon, and M. Neerincx, “Driver readiness model for regulating the transfer from automation to human control,” in *Proceedings of the 22nd international conference on intelligent user interfaces*, Mar. 2017, pp. 205–213.
4. S. Park and M. Whang, “Infrared camera-based non-contact measurement of brain activity from pupillary rhythms,” *Frontiers in physiology*, vol. 9, 2018.
5. B. Alhaji, J. Beecken, R. Ehlers, J. Gertheiss, F. Merz, J. Müller, M. Prilla, A. Rausch, A. Reinhardt, D. Reinhardt, C. Rembe, N. Rohweder, C. Schwindt, S. Westphal, and J. Zimmermann, “Engineering human-machine teams for trusted collaboration,” *submitted to: Big Data and Cognitive Computing*, 2020.
6. HerMes, https://www.simzentrum.de/en/hermes/.
7. M. Naber, G. A. Alvarez, and K. Nakayama, “Tracking the allocation of attention using human pupillary oscillations,” *Frontiers in Psychology*, vol. 4, 2013.
8. P. A. Barrionuevo, N. Nicandro, J. J. McAnany, A. J. Zele, P. Gamlin, and D. Cao, “Assessing Rod, Cone, and Melanopsin Contributions to Human Pupil Flicker Responses,” *Investigative Ophthalmology & Visual Science*, vol. 55, no. 2, pp. 719–727, Feb. 2014.
9. C. S. Herrmann, “Human EEG responses to 1-100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena,” *Experimental Brain Research*, vol. 137, no. 3-4, pp. 346–353, Apr. 2001.
10. D. Regan, *Human brain electrophysiology: evoked potentials and evoked magnetic fields in science and medicine*. New York: Elsevier, 1989.
11. B. Allison, I. Sugiarto, B. Graimann, and A. Gräser, “Display optimization in SSVEP BCIs,” in *Computer-Human Interaction*, 2008, pp. 2–5.
12. T. W. Picton, M. S. John, A. Dimitrijevic, and D. Purcell, “Human auditory steady-state responses: Respuestas auditivas de estado estable en humanos,” *International Journal of Audiology*, vol. 42, no. 4, pp. 177–219, Jan. 2003.
13. Tobii, https://www.tobii.com/.
14. T. R. Hayes and A. A. Petrov, “Mapping and correcting the influence of gaze position on pupil size measurements,” *Behavior research methods*, vol. 48, no. 2, pp. 510–527, Jun. 2016.

15. S.-C. Zhang and Z.-Q. Liu, “A robust, real-time ellipse detector,” *Pattern Recognition*, vol. 38, no. 2, pp. 273–287, Feb. 2005.

16. A. M. Fernandes, “Discussion on paper “A Robust Real-Time Ellipse Detector” by Zhang and Liu,” *Pattern Recognition*, vol. 44, no. 2, pp. 488–489, Feb. 2011.

An Image Processing Pipeline for Automated Packaging Structure Recognition

Laura Dörr, Felix Brandt, Martin Pouls, and Alexander Naumann

FZI Research Center for Information Technology,
Haid-und-Neu Str. 10-14, 76131 Karlsruhe

Abstract Dispatching and receiving logistics goods, as well as transportation itself, involve a high amount of manual efforts. The transported goods, including their packaging and labeling, need to be double-checked, verified or recognized at many supply chain network points. These processes hold automation potentials, which we aim to exploit using computer vision techniques. More precisely, we propose a cognitive system for the fully automated recognition of packaging structures for standardized logistics shipments based on single RGB images. Our contribution contains descriptions of a suitable system design and its evaluation on relevant real-world data. Further, we discuss our algorithmic choices.

Keywords Logistics, image processing, pattern recognition, convolutional neural networks, industrial applications

1 Introduction

In logistics supply chains, goods are transported along many different network points and require to be manually checked at each of these points. Such manual inspections often include not only unit identity but also completeness or packaging instruction compliance. In an aim to enable further automation of such inspection processes, we designed a system for automated image-based packaging structure recognition. Hereby, we define packaging structure recognition as the task of recognizing and analyzing logistics transport units and their building structure, allowing for inference of packaging types, number and arrangement. Fig. 1.1 illustrates this task.



Figure 1.1: Illustration of Packaging Structure Recognition. Transport unit side faces are illustrated in red, yellow lines indicate packaging unit rows and columns.

While numerous related image-based systems have been introduced by both dedicated start-ups and experienced vision and logistics companies, we are not aware of alternative solutions to the task of 3D packaging structure recognition for standardized logistics shipments from a single RGB image. The tackled tasks often include the detection of single packages or multiple package shipments and their dimensions. In many cases, individual packages or objects are recognized and counted, or logistics transport labels are found and read. For instance, a system by logivations [1] tackles a similar use-case of automated goods receive by detection and reading of logistics transport labels. Further, the solution is able to measure logistics units and count visible object instances. A method proposed by Fraunhofer IML [2] tackles the related problem of empties counting and tracking. Apart from solving slightly different tasks, many comparable systems use supplementary image and data acquisition means, e. g. multiple camera systems or additional sensors such as laser scanners or infrared technologies (e. g. [3], [4]). Aside from image based methods, the usage of non-visual sensors and information, like barcodes or RFID tags, is applicable to the problem at hand. However, such methods are often more expensive and still error-prone, as sensor ranges are limited and hardware requirements are enormous. Some of the obstacles regarding RFID technologies in logistics are discussed in [5]. At the same time, the hardware requirements for a image-based system like ours are minimal as we make no special assumptions regarding camera hardware.

We propose a solution for the task of packaging structure recognition based on single RGB images of applicable transport units, meeting certain necessary restrictions. The algorithm presented in this work was previously introduced and evaluated in [6]. Further, the logistics context and the use-cases we focus on are explained thoroughly in the before-mentioned publication. In this paper, we supplement our work by discussing the algorithmic choices and conducting further experiments and evaluations. More precisely, we analyze the task and define a series of sub-steps which, when combined together, are able to solve the task. For each of these tasks, we discuss input and expected output, requirements and evaluate our algorithmic approaches. We give reasons for the algorithmic choices we made, in some cases by evaluating different options on our own real-world use-case-specific data set.

2 Problem and Setting

In this section, we discuss the problem of packaging structure recognition more detailed and introduce clarify some logistics terms used throughout this work. Further, the setting in which the system was designed and tested is described, and necessary restrictions are explained.

2.1 Terms and Definitions

Packaging Unit. Packaging units are any containers used to transport goods along a logistics supply chain. Though made of various materials, these containers are often highly standardized (e. g. small load carrier system (KLT) [7]).

Base Pallet. This term is used to describe the base unit on which logistics goods can be stacked for transport. A wide range of mostly standardized pallets exist (e. g. EPAL Euro Pallet [8]).

Transport Unit. A logistics transport unit refers to fully-packed, shipping ready assortment of goods. Usually, such a unit consists of one base pallet, one or multiple packaging units and additional optional components, for instance lids, security straps or transparent foils. When speaking of uniformly packed transport units, we refer

to transport units containing only one single type of packaging units of identical size. By regularly packed transport units, we mean units with a fully regular packaging pattern, i. e. all rows, columns and layers of packages contain the same number of packaging units.

Transport Unit Side Face. We use the term transport unit side face to refer to that part of a transport unit which is visible when looking at it frontally from an arbitrary side with horizontal visual axis. Each logistics transport unit has exactly four such side faces. An occlusion-free image covering a whole transport unit can at most show two neighboring side faces of the transport unit.

2.2 Problem Formulation

The problem of packaging structure recognition as tackled in this paper is the task of localizing and inferring the packaging structure of one or multiple stacked transport units in a single RGB image. Hereby, the packaging structure consists of the number and type of packaging units, the arrangement of these units and, optionally, the type of base pallet present.

2.3 Setting and Restrictions

The task of packaging structure recognition as described above is not always solvable without making further assumptions on logistics components and imagery. Thus, we try to define a setting and reasonable restrictions to achieve feasibility.

Packaging Restrictions. First of all, only regularly and uniformly packed transport units are considered. This is necessary as the packaging structure of non-regularly packed transport units can in general not be inferred by observing a single image of that unit. Further, restricting the method to such units allows for improved robustness as not every individual packaging unit needs to be detected and identified. Instead, one can assume the rows and columns of each transport unit side to have the same number and types of packages, which the proposed algorithm does.

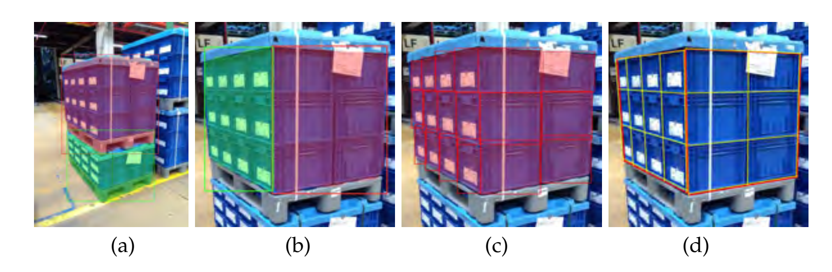
Imaging Restrictions. Additional restriction regarding image acquisition and contents. All images need to be taken in an upright orientation in such a way that vertical real-world structures (such as

transport unit edges) are roughly parallel to vertical image boundaries. Further, relevant transport units within the image are shown in their full extent and not occluded by any persons or objects. Additionally, we require transport units to be photographed in such an angle, that two of their side faces are clearly (and evenly well) visible.

Material Restrictions. For the time being, we limit our setting to a set of defined logistics components, i. e. packaging units and base ballets. As part of the algorithm relies on learning-based methods, we can only assume generalization to what is contained in the training data. Relevant packaging units in our setting are KLT packages and so-called tray packages, as is described more detailed in section 4.1.

3 Method Overview

This section contains a detailed description of the algorithm's coarse structure, i. e. we explain the series of independent tasks which build our image processing pipeline for packaging structure recognition. The four subsequent steps are illustrated in Fig. 3.1.



(a) (b) (c) (d)

Figure 3.1: Method Overview. (a) Step 1: Transport unit identification. (b) Step 2: Transport unit side face segmentation. (c) Step 3: Packaging unit identification and localization. (d) Step 4: Information consolidation, and output visualization.

Step 1: Transport unit identification and localization. The first step in our pipeline is to identify and localize all relevant logistics transport units in the input image. Relevant are such transport units which are visible in their full extend and without any occlusions. The expected output of this task is the number and locations of all

transport units within the image. Location information consists not only of the bounding box describing the image part fully covering the transport unit, but also a pixel-based instance mask, which provides valuable information for subsequent steps.

Step 2: Transport unit side face segmentation. The input of the pipeline's second processing step is a crop of the original image, containing exactly one, fully visible transport unit and the corresponding pixel mask. This step is performed for each transport unit found by the previous step. The expected output of this step are two segmentation masks for the transport unit's two visible side faces (see Section 2.3). Note that bounding boxes are again not sufficient, but more detailed, pixel-based information is required. The segmentation mask can be encoded as the coordinates of the four pixels showing the transport unit side face's four corners. As a transport unit side face can be described by a rectangle in 3D real space, it can be exactly localized by four pixel coordinates in our image, when assuming a distortion-free projective transform is underlying the image acquisition process.

Step 3: Packaging unit identification and localization. In this step, the packaging units for each transport unit side face are localized and classified. The task's input is the cropped handling unit image (same as input to step 2) and the corresponding handling unit side face information (output of step 2). The expected output is pixel-based information of the packaging unit's contained in each transport unit side. As in the previous step, this information can be encoded as the coordinates of four pixels for each packaging unit found within the image.

Step 4: Information consolidation. The last step used the information derived in the three previous steps to compose the desired output: the packaging structure of each transport unit contained and fully visible within the image. The most essential part of this task is the packaging number calculation for each transport unit side face. Here, the average width and height of each packaging unit is computed, considering the provided pixel-based segmentation information to account for varying object sizes due to perspective distortions.

4 Experiments and Implementation

4.1 Data Set



Figure 4.1: Example images from our data set. The left two images contain transport units with KLT packaging units, the right two images show transport units with tray packaging units.

A specific data set of 1267 images was acquired in a German plant of the automotive sector. All images comply to the restriction and assumptions described in section 2.3. As relevant for the setting in consideration, two different types of packaging units are present in the images: KLT packages and tray packages. Each image contains one single or multiple stacked transport units, which were thoroughly annotated, i. e. transport units, side faces, packaging units and base pallet are labeled on pixel basis. Of these 1267 images, 163 images are marked as dedicated evaluation data. The other images may be used for algorithm development, fitting and training purposes.

4.2 Transport Unit Segmentation

The step of transport unit segmentation is performed using a convolutional neural network (CNN) for instance segmentation. Namely, a Mask R-CNN [9] architecture with a Inception-v2 [10] feature extractor was trained using tensorflow 1.14 and the tensorflow object detection API. The model was pre-trained on the COCO object detection data set [11] and fine-tuned for the single-class task of recognizing and segmenting fully visible logistics transports units. Evaluation of the CNN and the transport unit segmentation step can be found in [6].

4.3 Transport Unit Side Segmentation

Two different approaches for the task of transport unit side segmentation are considered: One approach is based on machine learning, the other one employs classic image processing techniques.

First Approach: CNN. The first approaches uses a CNN for instance segmentation of analogous architecture to section 4.2. Using 75% the dataset's 1104 labeled training and validation images, the model was trained to recognize transport unit side faces, which are thoroughly labeled by four corner points each. The model achieved an mean average precision (mAP) of 0.877 on validation data (25% of the training images which were not used in model training) and 0.892 on the 163 evaluation images. Hereby, the mean average precision was computed in accordance to the COCO object detection's metric, i. e. as the averaged precision values at different intersection over union (IoU) thresholds of 0.5 to 0.95. To achieve the desired output format, a post-processing step, which fits four corner points to the instance segmentation mask found by the CNN model, is used. Hereby, an optimization problem choosing pixel coordinates for the corner points maximizing the region overlap with the CNN output mask, is solved. For more details, see [6].

Second Approach: Image Processing. Secondly, an image processing approach based on the Hough transform [12], a well-established method for detecting straight lines in images, was implemented. As package and transport units are of regular, rectangular shapes, an image crop showing one transport unit contains many linear structures. The approach's objective is to detect these linear structures, especially the edges of packaging units, to determine the transport unit side regions within the image. This is done in the following steps:

1. Line Detection: To detect qualifying horizontal and vertical structures, two different edge detection filter kernels are applied to the image, and the resulting edge images are binarized. Thereby, the image foreground is restricted to the actual transport unit region using the pixel mask which is input to the step of transport unit side segmentation. The binary images are used as input for the Hough transform in order to find linear structures. The line detection results are illustrated in Fig. 4.2 (a) and (b).

2. Vanishing Point Estimation: After the line detection has been performed, we try to determine the image's vanishing points [13] for vertical lines and for the visible transport unit sides' horizontal lines. To do so, we use a heuristic approach exploiting the knowledge on the image's contents and its geometric properties. We assume that the majority of vertical line segments detected correspond to vertical edges of the transport unit. After computing all intersection points of these vertical lines, we use the mean value of all intersection points as first guess for the unit's vertical vanishing point. Iteratively, we drop lines which do not get sufficiently close to the current vanishing point estimate. Then, we refine the estimate based on the intersection points of the reduced set of lines. This step is repeated several times with decreasing distance thresholds to obtain the final vanishing point position. For the two vanishing points of horizontal lines on our transport unit sides, we proceed similarly. We first try to find two accumulation points of horizontal line intersections: One on the left-hand side of the image and one on the right-hand side. Once again, we repeatedly assign lines in the vicinity of the vanishing points to its line set and use the reduced sets of lines to refine the vanishing point positions. Fig. 4.2 (c) illustrates vanishing point estimation and line assignments.

3. Side Boundary Estimation: Based on the vanishing points and corresponding lines, we try to segment the transport unit sides. To do so, start and end points for all horizontal line segments are determined by matching the line coordinates back to the binary edge image which we used before as input to the Hough operator. Using the obtained line endpoints, we estimate the transport unit side boundaries by fitting regression lines through corresponding endpoints of each line set and the vanishing point of the side's orthogonal lines. For instance, to find the left boundary of the left transport unit side, we regress a line through the vertical vanishing point and the topmost endpoints of all lines assigned to that vanishing point. The transport unit side corner points are inferred by intersecting these boundary lines. This is illustrated in Fig. 4.2 (d)-(f).

Overall, there is a considerable number of thresholds and similar parameters contained in this approach. For instance, in finding binary edge images, parameters involved are kernels sizes and patterns and binarization threshold. Further parameters are required

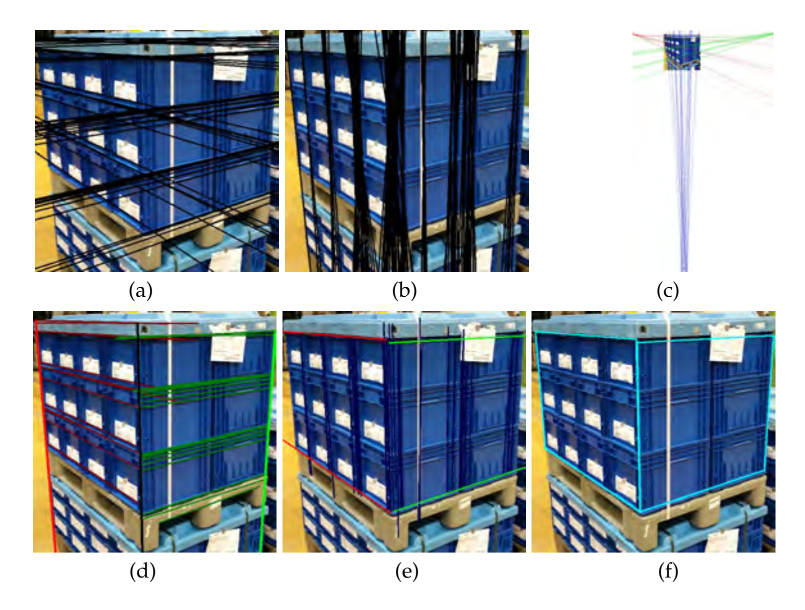


Figure 4.2: Segmentation of transport unit sides. Detected (a) horizontal and (b) vertical line segments, and (c) vanishing points of transport unit sides. (d), (e) Determination of boundary lines. (f) Resulting transport unit sides.

when performing the Hough transform, e. g. the minimum length of line segments to consider, as well as distance and angle resolutions. Also, in vanishing point estimation and line assignments, and in line endpoint determination, numerous threshold parameters are involved.

Evaluation. To evaluate the complete task of transport unit side segmentation, two different values are considered. First of all, the average intersection over union (IoU) for all transport unit sides is computed. Additionally, assuming sufficient accuracy to be given at an IoU of at least 0.8, the number of transport unit sides detected correctly is calculated. The results for both methods in consideration are shown in Table 1. The values show that, in the current implementation, the CNN outperforms our image processing approach by a great margin.

Table 1: Transport unit side segmentation evaluation results.

Method	Average IoU	Accuracy
CNN	0.8962	0.9029
Image Processing	0.6346	0.3006

Even though it is possible to tune the image processing algorithm to deliver precise results for single images or groups of images, we were not successful in finding parameters yielding good results on the whole data set. The evaluation values shown are the best values achieved by systematically varying the involved parameters in grid-search-like fashion. The CNN, on the other hand, easily generalizes to data as diverse as ours, due to the huge number of learnable parameters. Thus, the learning based algorithm appears superior, if not willing to distinguish different groups of images (e. g. by packaging type or size).

4.4 Packaging Unit Segmentation

For packaging unit segmentation, a CNN model analogously to section 4.2, is used. The model performs significantly better on KLT units compared to tray units, which is visible in the per class precision values (0.76 for KLT units compared to 0.67 for tray units), and in the overall error values for the different packaging types (see [6] for details).

First experiments applying image processing operations suggest that their application might be beneficial, especially in the case of tray packaging units. Package number determination could be tackled by analysing distances and frequencies of detected line segments in a rectified version of the transport unit side image. We plan to investigate this and conduct detailed experiments on that behalf in the future.

4.5 Pipeline Evaluation

In an end-to-end evaluation, the CNN-based packaging structure recognition pipeline achieved an overall accuracy value of approximately 84%. The metric applied was a custom, use-case specific

metric measuring the average ratio of correctly recognized and analyzed transport units per image. Again, more details can be found in [6].

5 Summary

We presented the problem of packaging structure recognition from single RGB images. For a specific logistics setting, we formulated reasonable restrictions and assumptions, and designed and presented a solution approach for this setting. The multi-step image processing pipeline was discussed and evaluated step by step, on our own use-case specific data set. Specifically for the step of transport unit side recognition, two different algorithms were implemented and compared systematically: A learning-based CNN for instance segmentation and a classic computer vision approach based on edge detection. The first outperformed the latter by a significant margin, which can be accounted for by the high variance in our image set and the CNN's superior generalization ability due to the higher number of parameters.

References

1. Logivations, "Ai-based identification solutions in logistics," https://www.logivations.com/en/solutions/agv/camera_identification.php, accessed: 2020-09-28.
2. J. Hinxlage and J. Möller, "Ladungsträgerzahlung per smartphone," *Jahresbericht Fraunhofer IML 2018*, pp. 72–73, 2018.
3. Vitronic, "Warehouse & distribution logistics," https://www.vitronic.com/en-us/logistics/warehouse-and-distribution, accessed: 2020-09-28.
4. Cognex, "Logistics Industry Solutions," https://www.cognex.com/industries/logistics, accessed: 2020-09-28.
5. M. K. Lim, W. Bahr, and S. C. Leung, "Rfid in the warehouse: A literature analysis (1995–2010) of its applications, benefits, challenges and future trends," *International Journal of Production Economics*, vol. 145, no. 1, pp. 409–430, 2013.

6. L. Dörr, F. Brandt, M. Pouls, and A. Naumann, "Fully-automated packaging structure recognition in logistics environments," *arXiv preprint arXiv:2008.04620*, 2020.
7. German Association of the Automotive Industry, "VDA 4500 - Small Load Carrier (SLC) System (KLT)," https://www.vda.de/en/services/Publications/small-load-carrier-(slc)-system-(klt).html, accessed: 2020-09-28.
8. European Pallet Association e.V. (EPAL), "EPAL Euro Pallet (EPAL 1)," https://www.epal-pallets.org/eu-en/load-carriers/epal-euro-pallet/, accessed: 2020-09-28.
9. K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
10. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
11. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.
12. P. V. Hough, "Method and means for recognizing complex patterns," Dec. 18 1962, uS Patent 3,069,654.
13. S. T. Barnard, "Interpreting perspective images," *Artificial intelligence*, vol. 21, no. 4, pp. 435–462, 1983.

Leistungsstarke und effiziente Bildinterpolation

Bastian Erdnüß[1,2] und Thomas Müller[1]

[1] Fraunhofer IOSB (Institute of Optronics,
System Technologies and Image Exploitation),
Fraunhoferstr. 1, 76131 Karlsruhe, Germany
[2] Institut of Photogrammetry and Remote Sensing (IPF),
Karlsruher Institut of Technology (KIT), 76128 Karlsruhe, Germany

Zusammenfassung Bildinterpolation ist Teil vieler Algorithmen zur Bildverarbeitung. Bei der Wahl der Methode wird meist ein Kompromiss zwischen Laufzeit und Qualität getroffen, der oft zu ungünstig für die Bildqualität ist. Ziel ist es, ein schnelles Verfahren mit hoher Qualität auf aktueller Hardware zu finden.

Keywords Bikubische Bildinterpolation, Lánczos-Interpolation

1 Einleitung

In dieser Arbeit werden verschiedene Interpolationsverfahren hinsichtlich ihrer Laufzeit und Ergebnisqualität untersucht. Die Interpolation ist wichtiger Bestandteil zahlreicher Bildverarbeitungsverfahren wie Bildreferenzierung und -mosaikierung. Bei der Verarbeitung von Live-Videodaten können dabei hohe Anforderungen an die Laufzeit der Interpolation bestehen. Grafikkarten etwa bieten häufig eine hochoptimierte bilinare Interpolation in ihren Texture Units an. Wenn die Qualität jedoch für eine Aufgabenstellung nicht ausreicht, muss auf rechenzeitintensivere Verfahren zurückgegriffen werden. Interpolationsverfahren werden zwar seit Jahrzehnten ausführlich untersucht, jedoch ändert sich mit der Weiterentwicklung der Hardware und der Verbreitung neuer Videostan-

dards (hinsichtlich Auflösung und Framerate) fortlaufend der Kompromiss, der zwischen Qualität und Laufzeit der Interpolationsverfahren getroffen werden kann.

Im Folgenden werden Interpolationsverfahren untersucht, die prinzipiell für die Echtzeitanwendung geeignet sind. Aus diesem Grund beschränkt sich dieser Artikel auf die Bildinterpolation mittels linearer separabler Filter mit beschränkten Trägern. Die Laufzeiten der Verfahren werden auf CPU und GPU ermittelt. Die Interpolationen werden auf verschiedene Testbilder mit häufig anzutreffenden Störungen angewandt wie Aliasing, Kompressionsartefakte und Farbrauschen. Um die Qualität der Verfahren zu vergleichen, wird die Interpolation iterativ mehrfach angewandt und die Degradierung der Bildqualität gegenüber dem Originalbild betrachtet.

2 Grundlagen

Ein Computerbild ist ein zweidimensionales $M \times N$-Tableau $I_{i,j} \in [0,1], i \in \{1,2,\ldots,M\}, j \in \{1,2,\ldots,N\}$ von Grauwerten. Die Grauwerte sind hier auf das Intervall $[0,1]$ normiert, wobei 0 schwarz und 1 weiß darstellt. Bei Farbbildern werden die drei Farbkanäle unabhängig voneinander behandelt und das Bild wird interpoliert, als ob es sich um drei Grauwertbilder handeln würde. Daher wird zur Vereinfachung im Folgenden von Grauwertbildern ausgegangen. Ziel der Bildinterpolation ist es, eine Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ anzugeben, die $I_{i,j}$ interpoliert, für die also

$$f(i,j) = I_{i,j} \tag{2.1}$$

für $i \in \{1,\ldots,M\}$ und $j \in \{1,\ldots,N\}$ gilt. Dadurch ist es möglich, die Grauwerte für das Bild nicht nur im Pixelraster $(i,j) \in \mathbb{N}^2$ sondern an beliebigen Zwischenstellen $(x,y) \in \mathbb{R}^2$ anzugeben.

Es gibt einige weitere Eigenschaften neben der Interpolationsbedingung (2.1), die ein Interpolationsverfahren haben kann oder die wünschenswert sind. Hier beschränken wir uns auf lineare Interpolationsverfahren, die separabel und lokal sind. Lokal bedeutet, dass zur Interpolation nur die Bildwerte in einer kleinen Umgebung des zu interpolierenden Punktes benötigt werden, und separabel bedeutet, dass die 2D-Bildinterpolation durch mehrmalige Anwendung

von 1D-Interpolationen auf den Bildzeilen und deren Ergebnissen entlang der Spalten gefunden werden kann (siehe Abb. 2.1).

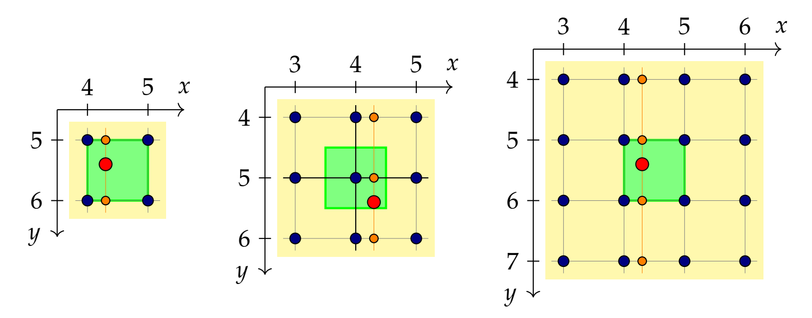


Abbildung 2.1: Separierbare Interpolation auf (4.3, 5.4) am Beispiel einer 2×2-, einer 3×3- und einer 4×4-Umgebung. Zunächst wird entlang der Zeilen auf die x-Koordinate 4.3 interpoliert (orangene Punkte). Anschließend werden diese entlang der Spalte (orangene Linie) auf die y-Koordinate 5.4 interpoliert (roter Punkt, Ergebnis der Interpolation). Alle interpolierten Punkte im grünen Quadrat hängen jeweils von denselben Bildpunkten $I_{i,j}$ der Umgebung ab, lediglich in unterschiedlichen Gewichtungen.

3 Methode

Folgende Interpolationstypen werden betrachtet:

- Nearest-Neighbour-Interpolation [1],
- bilineare Interpolation [1,2],
- bikubische Interpolation [1,2] (in verschiedenen Varianten),
- Lánczos-Interpolation [1,3] (mit verschiedenen Größen).

Die ersten beiden Varianten dienen als Referenz für die schnellsten und die letzte Variante als Referenz für die qualitativ besten Interpolationsmethoden. Durch Wahl geeigneter Bildgradienten zur Berechnung der bikubischen Interpolation wird versucht eine Methode zu finden, die ähnlich gute Ergebnisse wie die Lánczos-Interpolation liefert bei gleichzeitig geringerer Rechenzeit.

3.1 Interpolationsverfahren

Beim Nearest-Neighbour-Verfahren wird der Wert $f(x,y)$ durch den Farbwert $I_{i,j}$ des nähestgelegenen Pixels (i,j) mit $i = [x]$ und $j = [y]$ interpoliert, wobei $[\cdot]$ der Rundungsoperator ist. Das Bild wird gewissermaßen mittels Treppenstufen durch die Pixel interpoliert.

Bei der bilinearen Interpolation werden in den beiden Zeilen $j = \lfloor y \rfloor$ und $j = \lfloor y \rfloor + 1$ Zwischenwerte $f_j(x)$ berechnet und aus diesen $f(x,y)$ gebildet. Die Formeln dazu sind

$$f_j(x) = (i+1-x)I_{i,j} + (x-i)I_{i+1,j} \ , \tag{3.1}$$

$$f(x,y) = (j+1-y)f_j(x) + (y-j)f_{j+1}(x) \tag{3.2}$$

mit $i = \lfloor x \rfloor$ und dem Abrundungsoperator $\lfloor \cdot \rfloor$. Zwischen benachbarten Punkten wird dabei auf deren Verbindungsgeraden interpoliert.

Bei der Lánczos-Interpolation werden die Bildzeilen mit dem Lánczos-Kernel

$$l_a(x) = \begin{cases} \operatorname{sinc}(\pi x)\operatorname{sinc}(\frac{\pi x}{a}) & \text{für } |x| \leq a \\ 0 & \text{sonst} \end{cases} \tag{3.3}$$

gefaltet, wobei $a \in \mathbb{R}$ die Größe der betrachteten Umgebung bestimmt und $\operatorname{sinc}(x) = \frac{\sin x}{x}$ ist. D. h. es werden erst Zwischenwerte $f_j(x)$ und daraus $f(x,y)$ nach den Formeln

$$f_j(x) = \sum_i l_a(x-i)I_{i,j} \ , \tag{3.4}$$

$$f(x,y) = \sum_j l_a(y-j)f_j(x) \tag{3.5}$$

berechnet, wobei nur Summanden mit $i \in [x-a, x+a]$ und $j \in [y-a, y+a]$ einen Beitrag zur Summe liefern und $f_j(x)$ auch nur für entsprechende j berechnet werden muss.

Bei der bikubischen Interpolation wird zwischen zwei benachbarten Punkten mittels eines kubischen Polynoms interpoliert. Neben den beiden Randpunkten werden auch die Steigungen des Polynoms an den Rändern vorgegeben, wodurch es eindeutig bestimmt

ist. Sind G_k die Funktionswerte in einer Zeile und H_k deren Steigungen, so wird zwischen k und $k+1$ mit

$$g(x) = s^2(1+2t)G_k \;+\; t^2(1+2s)G_{k+1} \;+\; s^2tH_k \;-\; st^2H_{k+1} \quad (3.6)$$

interpoliert, wobei $s = k+1-x$ und $t = x-k$ die beiden Abstände von $x \in [k,\, k+1]$ zu den umliegenden beiden Punkten sind. Um die Abhängigkeiten von den Daten explizit zu machen, wird $g(x)$ auch ausführlicher als $g(x,\, G_k,\, G_{k+1},\, H_k,\, H_{k+1})$ notiert.

Es bezeichne I^x das Gradientenbild von I in x-Richtung, I^y das Gradientenbild in y-Richtung und I^{xy} dessen gemischte zweite Ableitung, jeweils auf den ganzzahligen Pixeln (i,j) definiert. Analog zur bilinearen Interpolation werden zunächst in den beiden Zeilen $j = \lfloor y \rfloor$ und $j = \lfloor y \rfloor + 1$ Zwischenwerte $f_j(x)$ aus I und I^x berechnet. Um jedoch die Interpolation in y-Richtung durchführen zu können, müssen auch die y-Gradienten $f_j^y(x)$ aus I^y und I^{xy} nach x interpoliert werden. Anschließend kann $f(x,y)$ aus $f_j(x)$ und $f_j^y(x)$ in y-Richtung interpoliert werden. Die Formeln dazu sind

$$f_j(x) = g(x,\, I_{i,j},\, I_{i+1,j},\, I^x_{i,j},\, I^x_{i+1,j}) \;, \quad (3.7)$$

$$f_{j+1}(x) = g(x,\, I_{i,j+1},\, I_{i+1,j+1},\, I^x_{i,j+1},\, I^x_{i+1,j+1}) \;, \quad (3.8)$$

$$f^y_j(x) = g(x,\, I^y_{i,j},\, I^y_{i+1,j},\, I^{xy}_{i,j},\, I^{xy}_{i+1,j}) \;, \quad (3.9)$$

$$f^y_{j+1}(x) = g(x,\, I^y_{i,j+1},\, I^y_{i+1,j+1},\, I^{xy}_{i,j+1},\, I^{xy}_{i+1,j+1}) \;, \quad (3.10)$$

$$f(x,y) = g(y,\, f_j(x),\, f_{j+1}(x),\, f^y_j(x),\, f^y_{j+1}(x)) \;. \quad (3.11)$$

Die eigentliche Interpolation wird mit einer festen Anzahl an Rechenschritten und Speicherzugriffen ausgeführt. Die Qualität der Interpolation hängt jedoch von der Qualität der Gradientenbilder I^x, I^y und I^{xy} ab. Diese können mittels disketer Faltung erzeugt werden. Von der Kernelgröße dieser Faltung hängt die Gesamtlaufzeit der Interpolation ab. Da die Berechnung der Gradientenbilder jedoch im Pixelraster erfolgt, kann diese effizienter implementiert werden, als es bei der Interpolation der Fall ist. Denn Zwischenergebnisse können wiederverwendet werden, Vektorsierung kann besser genutzt werden und der Prozessor-Cache wird weniger belastet.

3.2 Diskrete Ableitungsoperatoren

Die Gradientenbilder I^x, I^y und I^{xy} werden wiefolgt aus dem Eingangsbild I berechnet. Im ersten Schritt wird I^x zeilenweise aus I berechnet und (ggf. parallel dazu) mit dem gleichen Verfahren I^y spaltenweise aus I. Anschließend wird ebenfalls mit dem gleichen Verfahren I^{xy} zeilenweise aus I^y berechnet. Daher ist es hier ausreichend, die Berechnung von I^x aus I zu behandeln.

Im einfachsten Fall wird I^x durch den Differenzenquozienten

$$I^x_{i,j} = \frac{I_{i+1,j} - I_{i-1,j}}{2} \tag{3.12}$$

approximiert. Das entspricht der Faltung mit dem schiefsymmetrischen Kernel $[1, 0, -1]/2 = [\frac{1}{2}, 0, -\frac{1}{2}]$ in x-Richtung und führt zur Standardvariante der bikubischen Interpolation, wie sie in den meisten Softwarebibliotheken implementiert ist. Andere Ableitungsoperatoren werden gebildet, indem mit größeren schiefsymmetrischen Kerneln gefaltet wird. Diese haben die Form $[A_n, \dots, A_1, 0, -A_1, \dots, -A_n]$ und ihre Anwendung kann effizient als

$$I^x_{i,j} = \sum_{k=1}^{n} A_k (I_{i+k,j} - I_{i-k,j}) \tag{3.13}$$

implementiert werden. Daher wird bei Ableitungskerneln hier nur die Hälfte der Koeffizienten $[A_1, \dots, A_n]$ aufgelistet und z. B. der Ableitungskernel zu Gleichung (3.12) verkürzt als $[1]/2 = [0.5]$ geschrieben.

Zusätzlich kann in y-Richtung auch noch geglättet werden. Wird z. B. beim Ableitungskernel $[0.5]$ in y-Richtung mit dem Glättungskernel $[1, 2, 1]/4$ gefaltet, erhält man den Sobel-Operator [1]. Zu allen Ableitungskerneln wurde auch eine Glättung in Querrichtung erprobt. Diese Varianten haben jedoch durchweg zu schlechteren Ergebnissen geführt (deutliche Tiefpassfilterwirkung im Ergebnisbild), sodass sie hier nicht weiter verfolgt werden.

Drei Klassen von Ableitungskerneln werden betrachtet, sie werden als Diff, OptDiff und LanczosDiff bezeichnet. Diff sind die klassischen Ableitungskernel, die die größte Konsistenz im Fourier-Raum

haben, wenn der Pixelabstand gegen 0 geht. OptDiff sind dagegen unter Berücksichtigung der diskreten Pixel über das gesamte Fourier-Spektrum optimiert. Die Koeffizienten für beide stammen aus den Tabellen B.1 und B.2 in [4], siehe Tabelle 1. Das rekursive Filter in Tabelle B.3 aus [4] wurde ebenfalls getestet, allerdings konnten damit keine guten Ergebnisse produziert werden.

Tabelle 1: Koeffizienten für Diff und OptDiff für verschiedene n gemäß [4]

n	**Diff**	**OptDiff**
2	[8, −1] / 12	[0.758, −0.129]
3	[45, −9, 1] / 60	[0.848, −0.246, 0.048]
4	[672, −168, 32, −3] / 840	[0.896, −0.315, 0.107, −0.0215]
5	[2100, −600, 150, −25, 2] / 2520	[0.924, −0.360, 0.152, −0.0533, 0.0109]

Das Ziel von LanczosDiff ist es, möglichst ähnlich zur Lánczos-Interpolation zu sein. Wird eine Bildzeile j mittels Lánczos-Interpolation zu $f_j(x) = \sum_i l_a(x-i) I_{i,j}$ interpoliert, kann daraus deren Ableitung $I^x_{k,j} = f'_j(k) = \sum_i l'_a(k-i) I_{i,j}$ an den Pixelpositionen berechnet werden, um das Gradientenbild I^x zu bilden. Das Ergebnis entspricht der diskreten Faltung mit dem schiefsymmetrisch fortgesetzten Kernel $[l'_a(1), l'_a(2), \ldots, l'_a(n)]$ mit $n = \lfloor a \rfloor$ und

$$l'_a(k) = \frac{(-1)^k}{k} \operatorname{sinc}\left(\frac{\pi k}{a}\right) \ . \tag{3.14}$$

4 Ergebnisse

Zur genauen Analyse und Verdeutlichung der Wirkung der unterschiedlichen Verfahren wird ein Bild mehrmals interpoliert, sodass Interpolationsfehler verstärkt und die Ergebnisse qualitativ gut unterscheidbar werden. Hierzu wird ein Eingabebild m Mal um das Bildzentrum mit dem Winkel $360°/m$ rotiert. Die sich ergebende Volldrehung um 360° wird anschließend mit dem Originalbild visuell verglichen. Dabei können zwar subjektive Eindrücke in die Bewertung einfließen, jedoch hat sich eine automatische Auswertung über Differenzbilder als nicht zielführend erwiesen, da diese vor allem einen Tiefpassfiltereffekt der Interpolation nicht angemessen




Abbildung 4.1: Testbild (links) und mittels Lánczos-Interpolation mit $a = 6$ berechnete, um 15° rotierte Variante (erste Rotation zu $m = 24$; rechts).

gewichtet. Ebenso ist eine Auswertung über das Fourierspektrum nicht zweckmäßig, da Realdaten etwa bei Konsumer-Sensoren Aliasing enthalten können.

Abb. 4.1 zeigt ein Eingabebild und veranschaulicht das Vorgehen. In Abb. 4.2 sind die Ergebnisse einiger Interpolationsverfahren an einem Ausschnitt dieses kontrastscharfen Eingabebildes für $m = 24$ dargestellt. Als Umgebungsgröße wird 12×12 verwendet (d. h. $a = 6$ bei der Lánczos-Interpolation und $n = 5$ bei allen Varianten der bikubischen Interpolation), was sich als günstig erwiesen hat. Wie man erkennen kann, erzeugt die Nearest-Neighbour-Interpolation zwar ein scharfes Ergebnis, ist aber bei feinen Strukturen nicht ortserhaltend. Wie erwartet ist die bilineare Interpolation am unschärfsten und die standard bikubische auch deutlich unscharf. OptDiff bietet diesbezüglich eine deutliche Verbesserung. Das schärfste Ergebnis mit der besten Detailerhaltung liefert die Lánczos-Interpolation. Bei genauer Betrachtung von Abb. 4.2 fällt jedoch auf, dass diese auch den stärksten Klingeleffekt an scharfen Farbkanten erzeugt, d. h. diese vervielfacht.

Das Diff-Ergebnis (ohne Abbildung) liegt bzgl. Detailtreue zwischen standard bikubisch und OptDiff. Das Ergebnis mit Lanczos-Diff (ohne Abbildung) ist ebenso gut wie OptDiff, nur geringfügig anders.

Wie die Rechenzeiten der Verfahren in Tabelle 2 zeigen, benötigen OptDiff und LanczosDiff knapp das Doppelte der Rechenzeit der standard bikubischen Interpolation, wenn man nur eine CPU zur Verfügung hat. In Anbetracht des signifikanten Bildqualitätgewinns

ist das durchaus akzeptabel. Lediglich Lánczos liefert noch bessere Ergebnisse, ist aber hinsichtlich Rechenzeit zumeist in der Praxis ungeeignet.

Ganz anders sieht es auf der GPU aus. Hier lässt sich Lánczos erstaunlich effizient realisieren. Die Berechnungen dauern nur unbedeutend länger als standard bikubisch bei wesentlich besseren Ergebnissen. OptDiff und LanczosDiff rechnen sogar etwas länger als Lánczos und sind weder hinsichtlich Ergebnis noch Rechenzeit eine echte Alternative dazu. Sie wären aber für die meisten Anwendungen schnell genug und erzielen auch bessere Ergebnisse als standard bikubisch.

Tabelle 2: Rechenzeiten der Interpolationsverfahren für ein Bild der Größe 960×640 Pixel auf einer Intel i7-4770 CPU mit 3.40 GHz bzw. einer NVidia GeForce 940MX GPU. Das Pluszeichen verdeutlicht die Verteilung auf Ableitungsberechnung und eigentliche Interpolation.

Interpolation	CPU	GPU
Nearest-Neighbour	24 ms	3.7 ms
bilinear	31 ms	4.1 ms
standard bikubisch	63 ms	11.9 ms
Diff	33 ms + 82 ms = 115 ms	20.0 ms + 4.4 ms = 24.4 ms
OptDiff	33 ms + 82 ms = 115 ms	19.9 ms + 4.4 ms = 24.3 ms
LanczosDiff	33 ms + 82 ms = 115 ms	18.1 ms + 4.4 ms = 22.5 ms
Lánczos	21 s	17.5 ms

Abschließend werden in Abb. 4.3 noch ein Bild mit starken MPG-Artefakten aus einem hochgradig komprimierten Video und in Abb. 4.4 ein Bildbeispiel mit starkem Farbrauschen betrachtet, um die Auswirkung von Störeinflüssen zu untersuchen. Wie man in beiden Abbildungen erkennen kann, ergeben sich die gleichen Aussagen hinsichtlich Bildqualität wie beim obigen Eingabebild, das bedeutet, die Verfahren erweisen sich als robust gegenüber solchen Störungen.

5 Zusammenfassung

Heutige GPUs erlauben eine hocheffiziente Anwendung der Lánczos-Interpolation, welche qualitativ bestmögliche Resultate mit

hoher Detailtreue liefert, und das bei einer ähnlichen Rechenzeit wie die standard bikubische Interpolation.

Auf der CPU ist die Lánczos-Interpolation in der Praxis meist zu langsam. Hier erzielen die OptDiff- und LanczosDiff-Interpolation hochqualitative Ergebnisse bei lediglich einer knapp doppelt so hohen Rechenzeit wie die standard bikubische Interpolation.

Literatur

1. *Wikipedia: The Free Encyclopedia*, (abgerufen: 30. September 2020). [Online]. Available: https://www.wikipedia.org
2. W. H. Press et al., *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. USA: Cambridge University Press, 1992.
3. W. Burger and M. J. Burge, *Principles of Digital Image Processing: Core Algorithms*, ser. Undergraduate Topics in Computer Science. Springer London, 2009.
4. H. Scharr, “Optimale Operatoren in der Digitalen Bildverarbeitung,” *Dissertation, Universitätsbibliothek Heidelberg*, 2000.

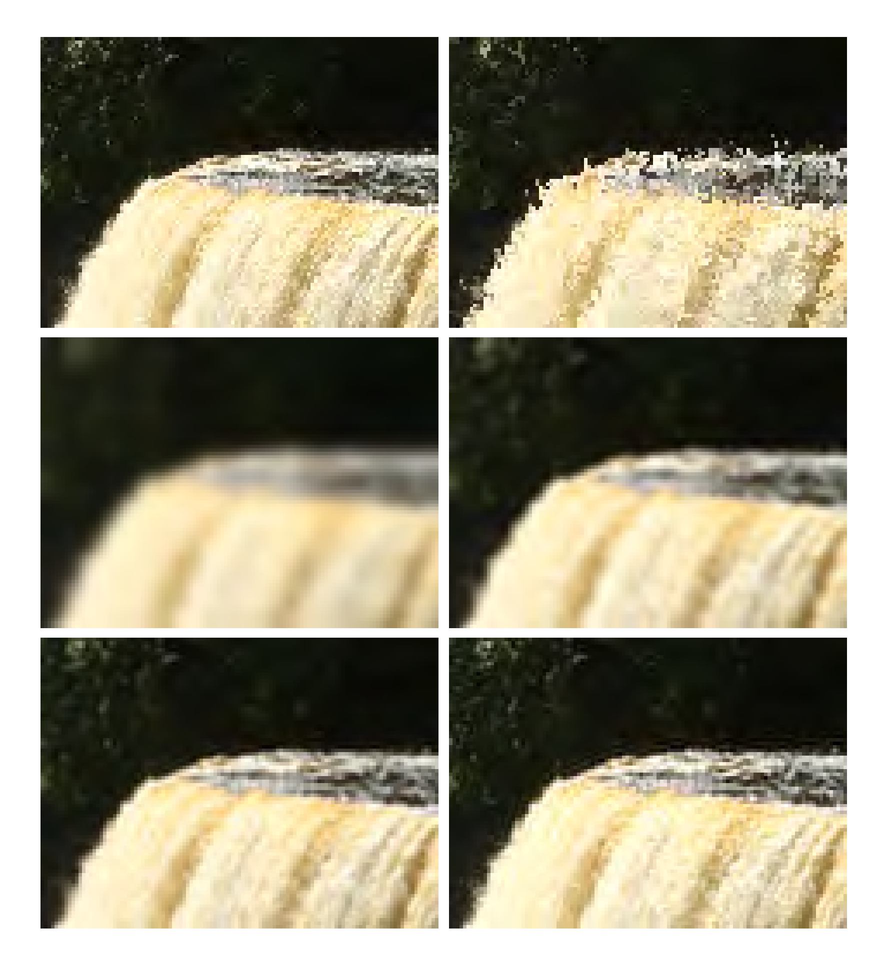


Abbildung 4.2: Ausschnitt des Testbildes aus Abb. 4.1 (oben links) sowie Interpolationen mit Nearest-Neighbour (oben rechts), bilinear (Mitte links), standard bikubisch (Mitte rechts), OptDiff (unten links) und Lánczos (unten rechts). Das Bild wurde jeweils $m = 24$ Mal rotiert.



Abbildung 4.3: Testbild (oben links) mit Ausschnittsvergrößerung (oben rechts) sowie Interpolationen bilinear (Mitte links), bikubisch (Mitte rechts), Opt-Diff (unten links) und Lánczos (unten rechts). Das Bild wurde jeweils $m = 24$ Mal rotiert.

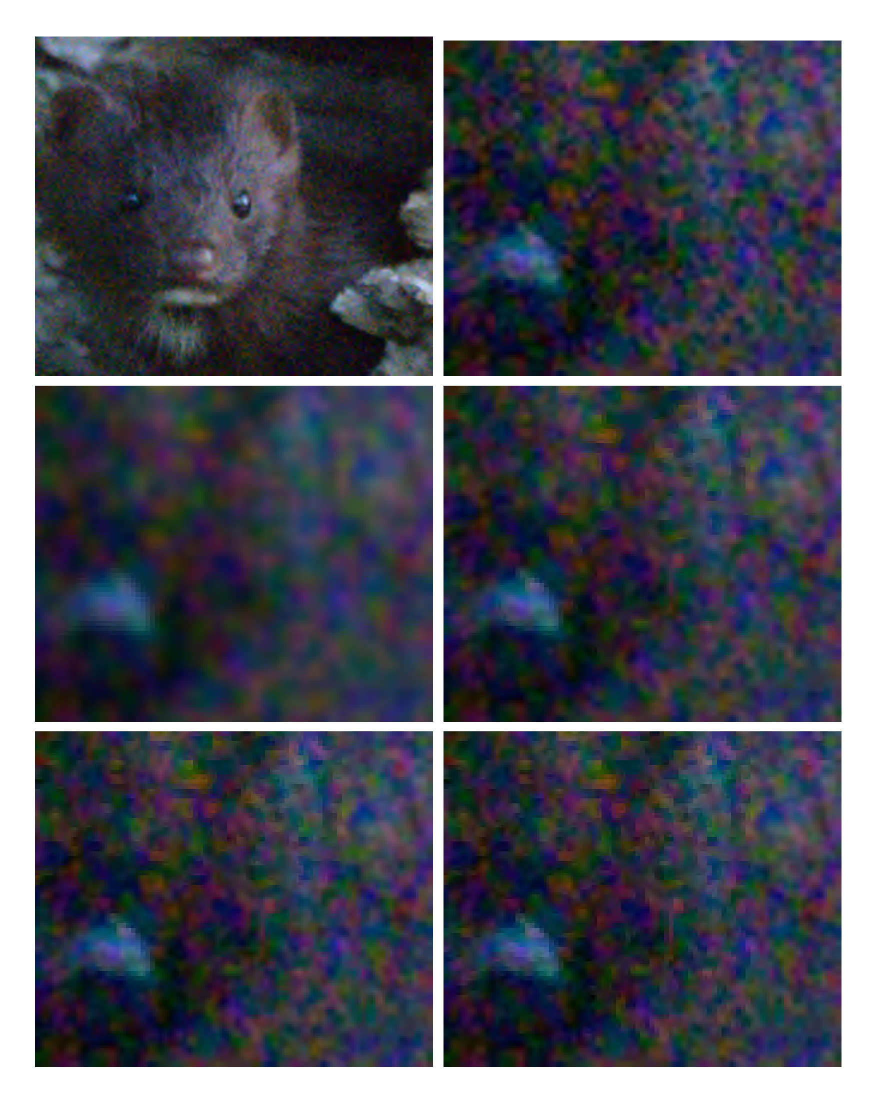


Abbildung 4.4: Testbild (oben links) mit Ausschnittsvergrößerung (oben rechts) sowie Interpolationen bilinear (Mitte links), bikubisch (Mitte rechts), Opt-Diff (unten links) und Lánczos (unten rechts). Das Bild wurde jeweils $m = 24$ Mal rotiert. [Quelle: Foto von Bill Maynard]

Multi-Seed Region Growing Algorithm for Medical Image Segmentation

Marco Gierlinger, Dinah Brandner, and Bernhard G. Zagar

Johannes Kepler University Linz, Institute for Measurement Technology,
Altenberger Straße 69, AT-4040 Linz

Abstract We present a heuristic approach to segment an image into multiple regions for subsequent feature extraction. The algorithm is based on region growing and allows parallel implementation by employing multiple seeds, that independently grow a region until all pixels of the image have been assigned. Seeds are homogeneously dispersed in pixel space and the growth of regions is controlled by prioritizing neighboring pixels via a bucket queue. The heuristic is based on histograms that are built up during growth to derive binary images for each seed. These binary images are weighted by additive image fusion. A simple preprocessing technique is applied to tune the algorithm's outcome. We explain how input parameters influence the algorithm's outcome and how practical solutions can be obtained.

Keywords Image segmentation, region growing, feature extraction, image registration, parallel implementation

1 Introduction

In medical diagnostics, deep learning methods [1] allow for an increase in both sensitivity and specificity of diagnostic results. As a drawback, however, to obtain accurate results they rely on massive training data, which typically is not available in the required annotated quality since it requires labor-intensive labeling by experts. An unsupervised technique called region growing might improve that situation by providing fully automated computer-aided segmentation and feature extraction [2]. Region growing is used very extensively in medical diagnostic applications [3] and it has shown to be

very useful, e. g. in the diagnosis of cardiac disease, or tumor volume segmentation [4]. It is an easy to implement and fast processing algorithm that is growing a region by comparing unassigned neighboring pixels to those already assigned to a growing region. It is, however, prone to so-called leakage. Without any special consideration or improvement to the algorithm, it tends to assign pixels also outside of a homogeneous region where borders are thinned out or interrupted due to noise or other artifacts. In this work, we address a noise-resistant and highly parallelizable technique, which can segment MRI volume data with a global view on the problem.

A further target is to design a tool for temporal analysis of image sequences by comparison of extracted features. A common issue in comparing two or more images is to register them, for example, at different instances of time, or when the sensor with respect to the patient is aligned differently. The task considered here tries to solve this issue by so-called image registration [5]. The idea of our work is to register different images by a set of extracted features based on region growing. However, this paper focuses on the region growing algorithm only and future work will cover the image registration part.

2 Related Work

Seeded Region Growing (SRG) by Adams and Bischof [6] is an effective and well-known image segmentation algorithm. SRG grows one or more regions, initially called seeds, that can be single-pixel-sized or a set of adjacent pixels. The algorithm grows these distinct regions due to some homogeneous criterion until all pixels are assigned a region. Formally, the sth seed grows region A_s for every $s \in \mathbb{N}$ where s is less than or equal to a user-defined number of seeds k. Let $\vec{p} \in \mathbb{N}_0^3 = (p_0, p_1, p_2)$ be a pixel, where (p_0, p_1) is its position in pixel space and let $I(\vec{p}) = p_2$ be its intensity value. Let $N(\vec{p})$ be the set of neighboring pixels of $\vec{p}$ in pixel space and $N(A) = \{r \in A \mid N(r) \setminus A \neq \varnothing\}$ the neighboring pixels of region A. During an iteration, a pixel $\vec{p}$ is assigned to the region A_s if

$$\vec{p} \in N(A_s) \land \delta(A_s, \vec{p}) = \min_{\forall i \in \mathbb{N} \land i \leq k \ \forall \vec{y} \in N(A_i)} \{\delta(A_i, \vec{y})\}, \tag{2.1}$$

where function δ is defined as:

$$\delta(A, \vec{p}) = \left| I(\vec{p}) - \underset{\vec{y} \in A}{\text{mean}} \{ I(\vec{y}) \} \right| \tag{2.2}$$

The number of iterations equals the number of pixels, i. e. the algorithm halts when all pixels are partitioned. The condition of Eqn. (2.1) may hold for multiple regions A_s and a single pixel $\vec{p}$, however, according to the authors of the SRG implementation, a pixel cannot be assigned to multiple regions during a single iteration. As a consequence, two inherent order dependencies may occur, which may lead to different segmentation results as discussed by Mehnert and Jackway [7].

Anyway, the presented algorithm employs independent seeds, which can grow fully in parallel without a rendezvous before every pixel has been visited. Also, in this approach, the mean intensity of a region is neglected and growth is promoted where two directly neighboring pixels meet the condition of some homogeneous criterion only.

In many region growing algorithms, k seeds typically grow k regions, and selecting a proper set of seed positions is a non-trivial task and crucial to the outcome. In our approach, we use one or more seeds but positions are homogeneously dispersed in pixel space and the number of seeds does not necessarily equal the number of extracted regions.

3 Algorithm

The following algorithm is described for n-bit grayscale images in $\mathbb{N}_0^{w \times h}$, however, adapting it for higher dimensions should be straightforward. Multiple seeds are employed and aligned as grid with a user-defined width u and height v with $u, v \in \mathbb{N}$, $u \leq w$ and $v \leq h$. For every $k \in \mathbb{N}_0$, with $k < uv$, a seed position vector $(x, y) \in \mathbb{N}_0^2$ is defined as:

$$x = \left\lfloor \frac{w}{u}(k \mod u) + \frac{w}{2u} \right\rfloor, y = \left\lfloor \frac{h}{v} \left\lfloor \frac{k}{u} \right\rfloor + \frac{h}{2v} \right\rfloor, \tag{3.1}$$

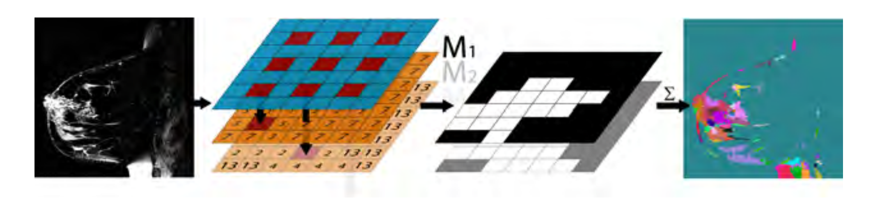


Figure 3.1: Region based segmentation applied to a breast MRI dataset [8].

as schematically depicted in red ($u = v = 3$) on the blue pane of Fig. 3.1. The algorithm walks through pixel space in an 8-adjacency flood-fill manner independently for each of the k seeds as follows: Let $\vec{p} \in \mathbb{N}_0^3 = (p_0, p_1, p_2)$ be a pixel, where (p_0, p_1) is its position in pixel space and let $I(\vec{p}) = p_2$ be its intensity value. A bucket queue is used to hold data objects $(\vec{p}, q)$, where $q \in \mathbb{N}_0$ with $q < 2^n$ is the bucket's index. Initially, the queue is filled with the seed only, which is a single pixel only. An iteration i is initiated by polling a bucket B_i. $\forall \vec{a} \in B_i \; \forall \vec{b} \in N'(\vec{a})$, a cost function $\delta_f : (\vec{a}, \vec{b}) \mapsto \{r \in \mathbb{N}_0 \mid r < 2^n\}$ is applied, where $N'(\vec{a})$ denotes all non-visited neighbors of $\vec{a}$. Cost function δ_f is defined as:

$$\delta_f(\vec{a}, \vec{b}) = \left| I(\vec{a}) - I(\vec{b}) \right| . \tag{3.2}$$

At the end of an iteration, each result is added to the queue as $(\vec{b}, \delta_f(\vec{a}, \vec{b}))$. A pixel counts as 'visited' when it is polled from (and not added to) the bucket queue.

3.1 Heuristic

Additionally, the number of newly assigned pixels m_i is tracked for each iteration i. A map $M_s \in \mathbb{N}_0^{w \times h}$, drawn as an orange pane in Fig. 3.1, is used for the sth of k seeds and every $r_{a_0,a_1} \in M_s$, where (a_0, a_1) is the position of $\vec{a}$, is set to $m_{\max}(i) = \max(m_0, m_1, ..., m_i)$ at iteration i. Finally, when the queue is empty, i. e. every pixel was visited, M_s is converted into a binary map by

$$r \in M_s = \begin{cases} 0 & r < m_{\max}(i) \\ 1 & \text{otherwise} \end{cases} \tag{3.3}$$

Then, all k binary maps are added up elementwise to $\sum_{s=1}^{k} M_s$, i. e. additive image fusion is applied, and normalized to the range from zero to $2^n - 1$ to suppress regions that occur less often than others. An example is highlighted in Fig. 3.1 (right) with random colors assigned for regions containing the same numerical value. The following pseudo code gives an additional overview of the algorithm for a single seed:

Algorithm 1: Generating binary Map M_s for a single seed.

```
1  add pixel p⃗ = (x, y, 0) to queue;
2  set every r ∈ M_s and m_max to zero;
3  while queue is not empty do
4  |   poll bucket B (with highest priority) from queue;
5  |   set m_i to zero;
6  |   for each a⃗ ∈ B do
7  |   |   if a⃗ hasn't been visited yet then
8  |   |   |   mark a⃗ as visited;
9  |   |   |   increment m_i by one;
10 |   |   |   set value at position of a⃗ in Map M_s to m_max;
11 |   |   |   for each b⃗ ∈ N'(a⃗) do
12 |   |   |       add (b⃗, f(a⃗, b⃗)) to queue ;
13 |   set m_max to max(m_max, m_i);
14 for each r ∈ M_s do
15 |   if r ≥ m_max then
16 |       set r to one;
```

In this algorithm, it is not necessary to visit every pixel. The iteration can halt when the number of non-visited pixels becomes smaller than $m_{\max}$. For the sake of simplicity, not every considered optimization is noted.

Growth is depicted in Fig. 3.2 and Fig. 3.3 for two arbitrary seed positions. Figure 3.2 shows a cumulative histogram of the pixel assimilation process and highlights the iteration for each seed where the largest peak is detected and Fig. 3.3 shows the corresponding growing process.

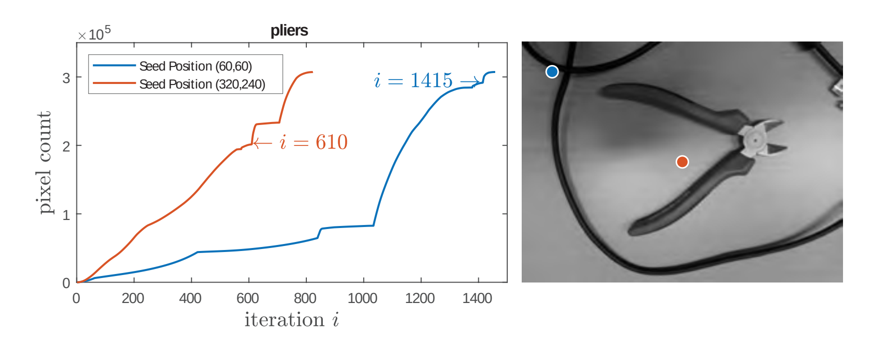


Figure 3.2: Left: Cumulative frequency of number of assigned pixels m_i on 'pliers' $(u, v) = (640, 480)$ and largest step $m_{\max}$ is found at highlighted iteration i. Right: Image with initial seed positions indicated.

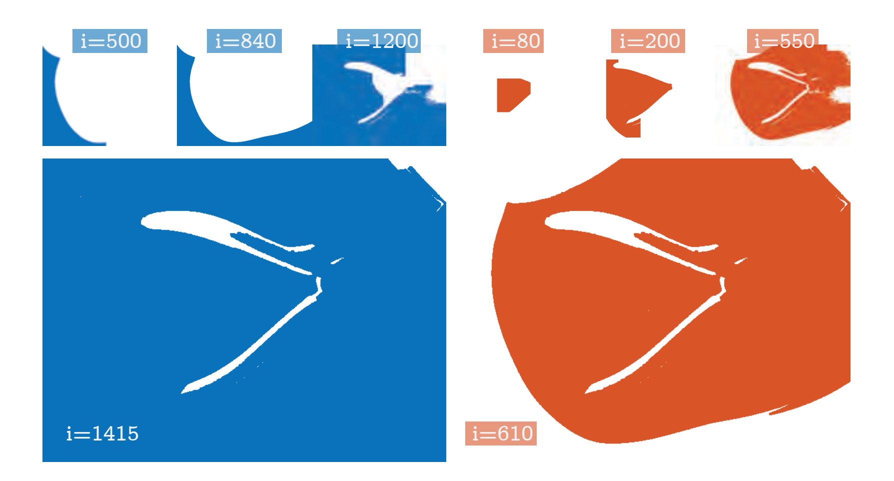


Figure 3.3: Growth of blue seed at $(60, 60)$ and orange seed at $(320, 240)$ on 'pliers'.

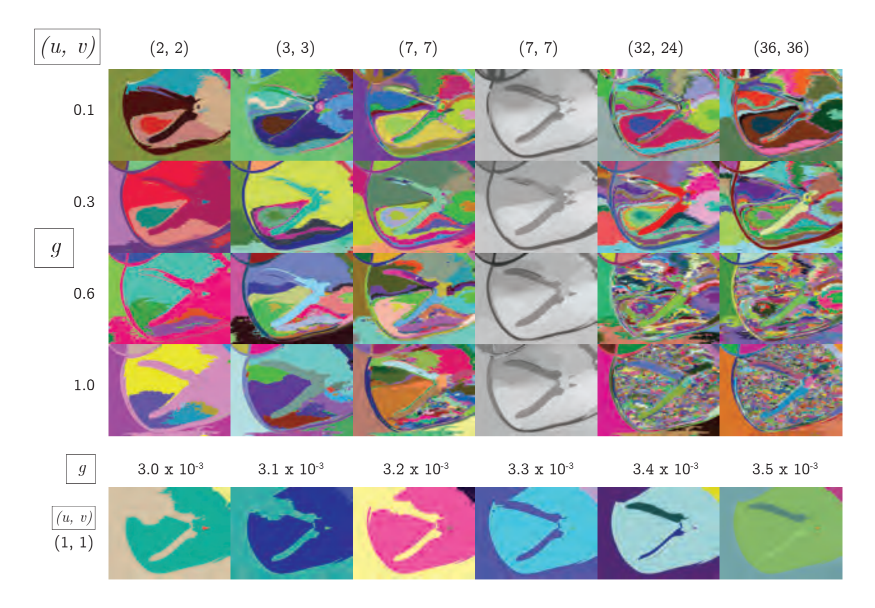


Figure 3.4: Tuning through the algorithm's solution space by adjusting the seed grid size uv and scaling parameter g. Random colors are assigned to distinctive regions. The fourth column from left contains the same regions as the third one and regions are colored by its mean intensity of the pixels of the original image.

3.2 Preprocessing

The algorithm's result is influenced by an input image and the selected grid size so far. However, it is desired to dynamically 'scan' for multiple acceptable solutions. Image quantization is used as a preprocessing step to decrease the image's bit depth, which has shown to be very useful to find practical solutions. Scaling parameter $g \in \mathbb{R}$ with $0 < g \leq 1$ is used to scale the image range to result in a lower bit depth. A user can tune the grid size uv, i. e. the density of dispersed seeds, and scaling parameter g as shown in Fig. 3.4 until a suitable solution is found. We may want to refer to [9], where image quantization is investigated when applied as a preprocessing step for dimensionality reduction in image classification pipelines.

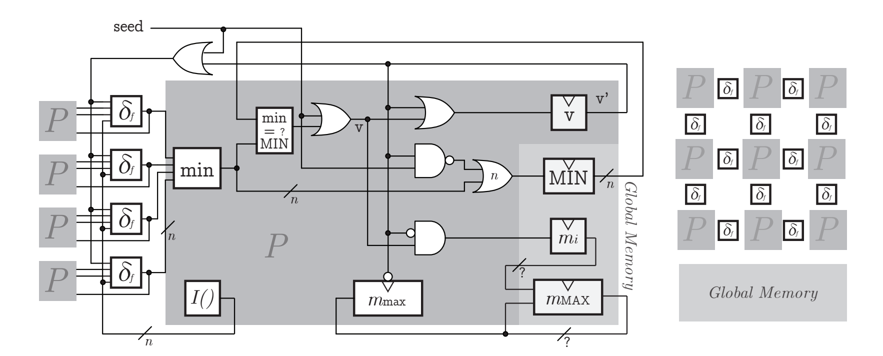


Figure 4.1: Sequential circuit representing a pixel in 4-adjacency. Left: Pixel in detail. Right: Pixel grid overview.

4 Parallel Implementation

This section intends to sketch the possibility of a highly parallelizable realization of the presented algorithm. While it is obvious to grow multiple seeds in parallel, it should also be noted, that per iteration, multiple pixels can potentially be examined in parallel as well. We present a concept for a sequential circuit, where the basic algorithm for the creation of a binary map is implemented such that it halts after as many clock cycles as iterations. The circuit is described for a pixel raster as schematically depicted on the right of Fig. 4.1, where $\boxed{\delta_f}$ denotes a connection in a 4-adjacent neighborhood. However, adapting it to 8-adjacency or 3D-connected cubes is straightforward. The single-bit input 'seed' (upper left of Fig. 4.1) is set HIGH for the seed pixel to initiate the algorithm and all essential $\boxed{\text{blocks}}$ in Fig. 4.1 are implemented as:

- $\boxed{I()}$ outputs the pixel's intensity, which can be an unsigned integer between zero and $(2^n - 2)$, where n is the bit depth. The word $(2^n - 1)$ is defined as NULL within this section's context.
- $\boxed{\text{v}}$ denotes single-bit memory, which saves the state whether a pixel was visited or not. It is set when the global $\boxed{\text{MIN}}$ and

the local min become equal and stays high until the algorithm halts.

- δ_f calculates δ_f of two neighboring pixels as in Eqn. (3.2), if the outputs v of these two pixels differ from each other, i. e. one pixel is part of the region and one is a neighbor of it.
- min propagates the minimum of the four n-bit neighboring δ_f-results to the global MIN if the pixel was already visited or if it is a seed, otherwise NULL is sent to MIN.
- MIN finds the minimum of each's pixel min output
- $m_{\max}$ updates its value by the global m_{MAX} output until the pixel was visited.
- m_i is a parallel counter [10], which counts each pixel that is first-time visited. The result is compared to the previous value of m_{MAX} to determine and output the maximum of both values.

The performance bottleneck consists of the MIN and m_i blocks, where a clever design is required to keep propagation delays low. However, propagation delay, stray capacitance, and any other hardware related issues are neglected in this section and require further investigation. The '?'-bit width should be chosen to count at least the largest number of bordering pixels that may occur at a single iteration.

5 Results and Discussions

5.1 Tuning, Merge and Split

As seen in Fig. 3.4, the more seeds are employed the more the algorithm tends to oversegmentation. The fewer seeds are employed, i. e. the smaller a seed grid size is selected, the more the position of a single seed influences the overall outcome of the algorithm. If only

a few homogeneous regions are dominating the image, increasing the number of seeds will not necessarily increase oversegmentation. However, the algorithm might be re-executed with already extracted regions as input instead of the whole image to furtherly split them. Conversely, oversegmentation can be compensated for by merging adjacent regions, that have a similar mean intensity.

This paper intends to provide a low-level tool for high-level applications. Whether the solution space has optimum solutions or not, is an application-dependent consideration and requires some sort of 'oracle' or 'teacher' as known from active learning [11].

5.2 Performance

When scaling parameter g is decreased the algorithm's time complexity decreases as well. While the bit depth and the spatial image resolution influence the algorithm's time complexity, it is believed that the complexity will not necessarily increase as the number of dimensions does for the parallel implementation. Analogously, one might compare the growth of a square in 2D, a cube in 3D, or a tesseract in 4D, where each dimension has the same spatial resolution.

5.3 Application and Future Work

When regions are extracted, subsequently, our goal is to register a large set of regions of MRI breast cancer image sequences. Also, we would like to apply our algorithm to pairs of stereo images like the Middlebury Stereo Datasets [12] to investigate the possibility of stereo matching techniques. Further investigations will cover non-rigid shape registration methods and similarity measure of how well registered regions match.

6 Conclusions

Based on seeded region growing, an algorithm was designed to support feature extraction in the field of medical diagnostics, however, it is not necessarily limited to this type of image. While the presented

algorithm is similar to the common region growing algorithms, the used heuristic is a novel and potentially faster approach. There is no need to find specific seed positions but instead, it is required to scale a parameter to adjust the density of homogeneously dispersed seeds in pixel space. Another input parameter is applied in a preprocessing step to reduce dimensionality by image quantization. The combination of adjusting density and dimensionality was depicted to give an intuition of the usefulness for more application-oriented approaches where optimal solutions might be found by an oracle as known from active learning.

A sequential circuit was described in this paper to point out the possibility of a highly parallel implementation with low time complexity. Overall experimental results seem promising, however, further investigation is required to evaluate the quality of segmentation.

Acknowledgement

This work has been supported by the LCM K2 Center within the framework of the Austrian COMET-K2 programme.

References

1. M. Ulrich, P. Follmann, J.-H. Neudeck, "A comparison of shape-based matching with deep-learning-based object detection," *tm – Technisches Messen, 2019, vol. 86(11), pp. 685–698*, 2019.
2. R. Neubecker, M. Heizmann, "Practice-oriented procedures for evaluating classifying image processing systems," *tm – Technisches Messen 2018, 85(4), pp. 252–267*, 2018.
3. Chowdhary, C.L., Acharjya, D.P., "Segmentation and feature extraction in medical imaging: A systematic review," *Procedia Computer Science, 167, pp. 26-36.*, 2020.
4. N. Mitschke, M. Heizmann, "About the detectability of objects in images by quantized neural networks," *tm – Technisches Messen, 2019, vol. 86(11), pp. 651–660*, 2019.
5. Z. Barbara, J. Flusser, "Image registration methods: a survey," *Image Vis. Comput., 21 (11) (2003), pp. 977-1000, 10.1016/s0262-8856(03)00137-9*, 2003.

6. R. Adams, L. Bischof, “Seeded region growing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16 (6) (1994), pp. 641-647*, 1994.
7. A. Mehnert, P. Jackway, “An improved seeded region growing algorithm,” *Pattern Recognit. Lett., 18 (1997), pp. 1065-1071, 10.1016/S0167-8655(97)00131-1*, 1997.
8. David Newitt, Nola Hylton, on behalf of the I-SPY 1 Network and ACRIN 6657 Trial Team, “Multi-center breast dce-mri data and segmentations from patients in the i-spy 1/acrin 6657 trials.” *The Cancer Imaging Archive. http://doi.org/10.7937/K9/TCIA.2016.HdHpgJLK*, 2016.
9. Ponti M., Nazaré T.S., Thumé G.S., “Image quantization as a dimensionality reduction procedure in color and texture feature extraction,” *Neurocomputing, 173 (2016), pp. 385-396*, 2016.
10. E. Swartzlander, “Parallel counters,” *IEEE Transactions on Computers, vol. C-22, pp. 1021-1024*, 1973.
11. Burr Settles, “Active learning literature survey,” *Computer Sciences Technical Report 1648, University of Wisconsin–Madison*, 2009.
12. D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling, “High-resolution stereo datasets with subpixel-accurate ground truth,” *German Conference on Pattern Recognition (GCPR 2014), Münster, Germany*, 2014.

Measuring similarity of rendered and real image pairs using domain translation by employing Conditional Generative Adversarial Networks

Naveen Raj Datha[1] and Marcus Thiel[2]

[1] Fraunhofer Institute for Factory Operation and Automation IFF, Sandtorstraße 22, 39106 Magdeburg
[2] Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, Data & Knowledge Engineering Group, Universitätsplatz 2, 39106 Magdeburg

Abstract One way for the visual inspection of assemblies with many variants is to compare camera images with the corresponding rendered view of the CAD model. In this paper, we address the problem to decide whether there are significant differences between camera and rendered images, which signal an assembly error. Our approach uses a Conditional Generative Adversarial Network (CGAN) to translate the camera image to a rendered like one, followed by error detection by comparing the translated and rendered images.

Keywords Automated visual assembly inspection, CGAN, deep learning, quality assurance, human-machine systems

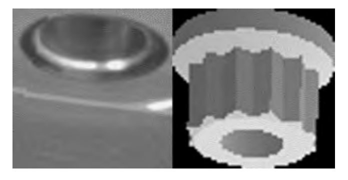
1 Introduction

This research is motivated by an inspection task in manual assembly. In order to ensure that a module is correctly assembled visual inspection is a frequent choice. When assembly errors may cause high costs in downstream processes or could lead to dangerous malfunction, an investment into a reliable automated inspection solution is of interest. For assemblies with many variants the following approach with cameras could be used. The cameras take images of





(a) Correct assembly



(b) Assembly error (missing part)

Figure 1.1: Pairs of camera and rendered images.

sufficient resolution of the parts to be inspected in such a way, that we know the position of the camera with respect to the assembly. This allows us to render the same view using the CAD model [1]. Then we compare the rendered view and the real camera image and to decide, whether there is an error. In the current work, we refer to the rendered view as the **CAD image** and the real camera image as **real-world image**.

The CAD model provides only geometrical information. A photo-realistic rendering is not possible. Yet, intensity changes can be expected, where surface normals change or neighbor pixels are on different objects or background. That is why the existing image processing approaches focus on the comparison of edges detected in CAD and real-world images [2]. As expected edges appear more or less distinct, and as there are further edges from texture and illumination, edge detection and comparison criteria need to be parametrized based on example images from the assembly. Whenever there are new parts, their finishing changes or lighting conditions are modified it may be necessary to adapt parameters and criteria again. Therefore, it is natural to ask whether there is a machine learning approach, which learns the classification of assembly errors and correctly mounted parts based on annotated example images with only few examples for assembly errors. In the current work, we introduce such a data-driven learning mechanism. Our approach can be easily adapted to new assembly products, parts or conditions, with minimal human expert involvement.

Dataset: The dataset we used consists of image pairs obtained from 42 real-world assemblies of 3 different assembly products. Figures 22.1(a) and 22.1(b) show such sample image pairs. From all the 42 experiments we have around 24333 inspection tasks, i. e. 24333 image pairs that can be used for training and testing. In the 24333 image pairs only 260 image pairs are assembly errors and all the other are correctly assembled samples. For our learning task, we leave out the samples from 9 experiments (3 from each assembly type) for testing and use the remaining samples for training. From here on, we refer to the correctly assembled samples as **negative samples** and the assembly error samples as **positive samples**.

We categorize the parts-of-interest in our inspection tasks into 10 different categories based on their visual appearance. Figure 1.2 shows one sample from each of these categories and their names. Also, we increase the dataset size by performing data-augmentation (discussed later in section 2). We adjust all the images to aspect ratio 1 : 1 for the sake of ease of training Convolutional Neural Networks.



Figure 1.2: Sample images of 10 different categories. Names in order from left: Air-Adapters, Bolts-1, Bolts-2, Bolts-3, Bolts-4, Mounting-Plates, Stiffeners, Swivel-Nuts, Vent-Tubes, Miscellaneous categories.

2 Method

Suppose that the real image could be translated from the domain of real textured images to the domain of rendered like images, then an image comparison could be used for classification. Similar images would represent the correctly mounted parts and differences in the images would signalize assembly errors. The intention is to simplify the classification to asking whether a similarity measure for two images is above or below some threshold. The learning comes in with domain translation. The advantage is that for learning the domain translation we only need negative samples.

Figure 2.1 shows the conceptual idea behind our methodology, the data flows from left to right in the pipeline. The first stage is about pre-processing data. We generate a mask from the CAD image, where the pixels of part of interest are assigned value 0 and the background pixels are assigned a value 1. We then extract the background from the real-world image by simply multiplying the real-world image with the mask image. The extracted background is then added to the CAD image resulting in a hybrid image. We use this hybrid image as our ground-truth for training and also for classification of assembly errors. Figure 2.2 shows an example of these data pre-processing steps. In section 3 we also discuss results of experiments, where we omitted this pre-processing and kept the black background.

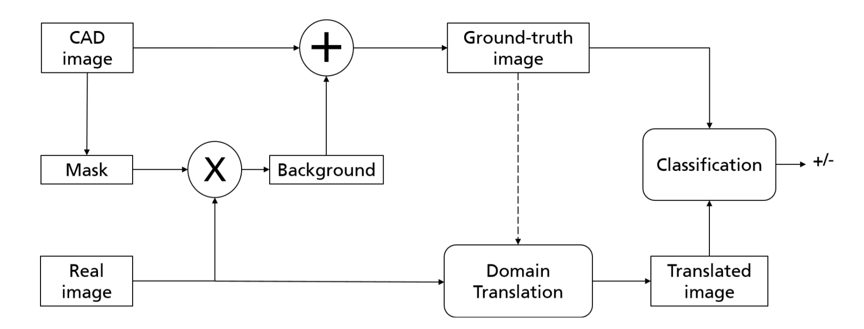


Figure 2.1: Conceptual diagram

The second stage is image domain translation. For domain translation, we choose a deep learning approach: Conditional Generative Adversarial Networks (CGAN) [3]. CGANs are a special case of Generative Adversarial Networks (GAN) [4] which are state-of-the-art image generation models. A CGAN consists of two blocks, generator and discriminator, both these blocks are made up of Convolutional Neural Networks. During training, the generator of CGAN learns to translate input real-world image to CAD domain. The discriminator on the other hand learns to distinguish between the generator's output and ground-truth CAD image, given the real-world image. For the generator network, we experimented with two different architec-

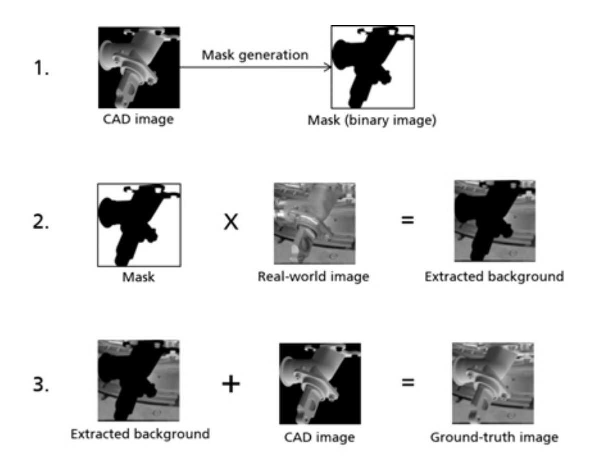


Figure 2.2: 3 step data pre-processing to obtain ground-truth images

tures from the state-of-the-art, the pix2pix [5] and U-Net [6] architectures. For the discriminator, we use the architecture proposed in pix2pix [5]. Also, to make domain translation invariant to Euclidean transformations or small changes of intensities, we apply data augmentation techniques such as Flipping, rotating, translating, small random increase/decrease of pixel values on the training data set. The third stage of the conceptual design consists of a classification block, where we compare the translated and ground-truth images to detect assembly errors. Note that, though we use the terminology of classification here, no learning process happens in this block. The actual learning process happens only in the domain translation block.

In our approach, we need image comparison metrics for two purposes. One for evaluating the quality of image translation, the other for comparing the translated and ground-truth images in the classification block. For the purpose of measuring the image translation quality we use the Structural SIMilarity (SSIM) Index [7]. Given a pair of perceptually similar images, SSIM gives a measure of similarity in the structural information of the images. A good domain translation model, while translating the domain of an input image,

should not affect/degrade the structural information present in it. Thus, comparing the structural information in the translated images with the structural information in the ground-truth images forms a good basis for testing the translation quality. The SSIM metric serves this purpose here. Note that, for training and testing the CGAN performance, we only need the negative samples from the dataset, we do not need the positive samples here.

For comparing the translated and ground-truth images in the classification block, we use the Mean squared error (MSE). MSE is calculated as the mean of squared pixel intensity differences between the given pair of images. MSE is calculated at pixel level and does not take into account the neighborhood relations. However, MSE highly penalizes large deviations in pixel intensities. This factor greatly helps in classifying the borderline positive samples, where the parts-of-interest in the image pair are mostly similar with only some small differences, see figure 22.1(b). To classify a sample as negative or positive based on MSE, we need a threshold. We choose the threshold as a trade-off between the sensitivity and specificity measures. The sensitivity and specificity measures are calculated as

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}, \tag{2.1}$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}. \tag{2.2}$$

We calculate the sensitivity and specificity over a range of different thresholds and finally choose the threshold where the sum of both measures is maximum. We use 60% of our test samples, both negative and positive for this purpose. Also, to have a balanced dataset for threshold calculation, we perform data-augmentation to generate artificial positive samples, by simply swapping the CAD image of a given image pair with some different CAD image.

3 Results

We performed a set of experiments to evaluate the performance of our pipeline with pix2pix and U-Net generator architectures. The architectures in all our experiments were trained using the Adam

optimizer [8] with a learning rate of 0.0002. In case of discriminator we used Mean squared error (MSE) as the loss function, whereas in case of generator we used a weighted sum of Mean squared error (MSE) and Mean absolute error (MAE) as suggested in [5].

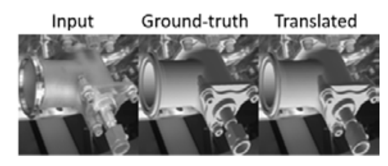
The results we report in this section were obtained for real-world to CAD image translation and by using images with background. Later in this section, we explain our observations on why translating CAD images to real-world results in poor translation quality and why it is not a good idea to use images without background for training. Also, the numbers reported in this section were obtained over original positive and negative test samples, no augmented data was included in these calculations.

Table 1: Image translation and classification results

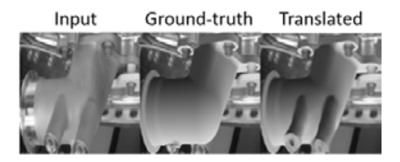
	Translation quality (Avg. SSIM)		**Classification performance (based on MSE measure)**	
Architecture	**Train-set**	**Test-set**	**Sensitivity**	**Specificity**
pix2pix	0.93	0.92	0.75	0.97
U-Net	0.94	0.93	0.85	0.98

Table 1 lists the results of the best performing pix2pix and U-Net generator models. The pix2pix generator model took relatively longer training time compared to U-Net for achieving similar translation quality. In both cases, image translation quality remains good and consistent over train and test sets, indicating that the models learned to generalize. Figures 22.1(a), 22.1(b), 22.1(c) show some sample image translation results obtained with the U-Net model. In terms of classification performance, although the specificity is almost the same in both cases, we achieved better sensitivity with U-Net translation. While the pix2pix pipeline misses to detect 25% error samples in the test set, U-Net performs slightly better by missing 15% defects.

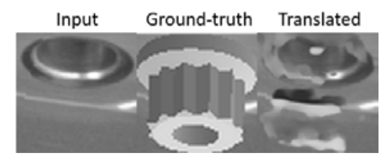
The results in table 1 were obtained using a common MSE threshold for all the 10 categories of objects. But, in production different types of errors can occur for different categories. For example, in our dataset, missing bolt is a most common error in case of bolts, whereas, in case of air-adapters the most common error is mount-



(a) Negative sample (No error)



(b) Positive sample (Wrong part)



(c) Positive sample (Missing part)

Figure 3.1: Image translation results

ing a wrong part. When the types of errors differ, the MSE values in these cases differ too and therefore it would be beneficial to use different cut-off values for each category of objects rather than using a common cut-off for all the categories. Table 2 summarizes the results we obtained after choosing individual cut-offs for each category. These results were obtained using the U-Net generator mentioned in table 1. Except for the Stiffeners category all other categories have sensitivity of 1.0 i. e., 100% error detection.

Stiffeners are a special category of objects which have an extreme aspect ratio, see figure 3.2. When these images are resized to a aspect ratio of 1 : 1, a lot of information about the part of interest is lost, and therefore it would be difficult to detect errors in such cases. We solved this problem by training the CGAN on image tiles, obtained by splitting each Stiffener image into multiple small tiles. And during classification, if any tile of an image is classified as positive then the image itself is classified as positive. Using this approach we achieved 100% error detection in case of Stiffeners too.

Translating CAD images to real-world: Figures 22.3(a) and 22.3(b) show the image translation results we obtained by training a CGAN model to translate CAD images to real-world images. The training

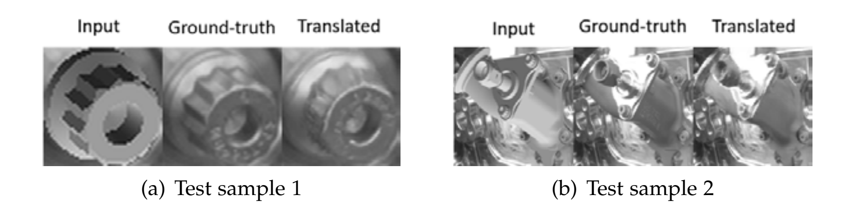
Table 2: Category-wise classification results obtained after choosing individual cut-offs for each category

Category	Sensitivity	Specificity
Air-Adapters	1.0	1.0
Bolts-1	1.0	0.99
Bolts-2	1.0	0.99
Bolts-3	1.0	0.98
Bolts-4	1.0	1.0
Mounting-Plates	1.0	0.97
Stiffeners	0.75	0.98
Swivel-Nuts	1.0	1.0
Vent-Tubes	1.0	1.0
Miscellaneous	1.0	0.96



Figure 3.2: Image of a Stiffener with its original aspect ratio (14 : 1)

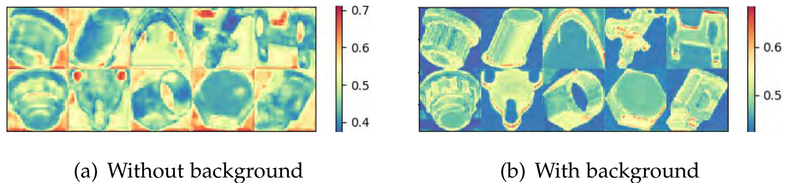
loss and the image quality stopped improving after we trained the model for a few hundred epochs. The possible reason here for poor image translation quality could be that, the process of transitioning from real-world to CAD-world is like a simplification process, the other way is not. Lets say there is a product which contains a part X, the part's CAD model image is X_c. Lets say for the purpose of training we obtained real-world images X_{r1}, X_{r2}, X_{r3} of this part when the product was assembled three different times. Now, when we train the CGAN model with X_{r1}, X_{r2}, X_{r3} as inputs and X_c as the common ground-truth for all three inputs, we are essentially training the model to simplify the inputs and converge the output to the pixel values seen in X_c. But, when we train the model with X_c as input and X_{r1}, X_{r2}, X_{r3} as ground-truths, the model learns to generate a mean output image that equally satisfies the constraints of all three ground-truths it has seen during training. Therefore, translating CAD images to real-world might always result in blurry outputs.



(a) Test sample 1 (b) Test sample 2

Figure 3.3: Blurry translation results obtained from a CGAN model trained to translate images from CAD domain to real-world domain

Using images without background: In the experiments where we trained the CGAN on images without background, we observed that the translation quality on training images was satisfactorily good, but the quality of outputs obtained on test-set was poor, indicating over-fitting. Figures 22.4(a) and 22.4(b) show the mean activation maps [9] we plotted for the top layers of the CGAN generator model to understand its behaviour. In figure 22.4(a), where we trained the model on images without background, we see that, in almost all categories of input images, the high activation values are in the background region (The reddish regions in the activation maps indicate high activations). But, in case of figure 22.4(b), where we trained the model on images with background, the activation values in the background region are lower than the part-of-interest, indicating that the model learned the structure of the part-of-interest. When a model is trained on images without background, the network might find it easier to learn about the black patches in the background, than learning about the complexity of structures in the region of interest. The Convolutional Neural Networks usually learn to find or extract the most common features that can differentiate one class from the other. Therefore in order to drive the network towards learning the right features for a given part of interest, we simply have to make sure that no other part or region in the image highly correlates with the part-of-interest. But this is not the case with black-background images. Whenever there is some specific part-of-interest in an image, there are always corresponding black-patches as well. Then the network might learn about the black-patches rather the part of interest. In short, the lower the correlation between background and



(a) Without background (b) With background

Figure 3.4: The activation maps of each category highlight the difference in behaviour of the CGAN model when trained on with and without background images. (Reddish regions in the images indicate higher activation values)

the region of interest, the better. The background added from the real-world image and the data-augmentation helps in achieving this randomness.

4 Summary

In manual assembly tasks, inspection of products assembled by humans is essential. One way of doing this is by automated visual inspection using camera images. Images captured in the real-world can be compared with images rendered from the CAD model to detect errors. The existing approaches focus on comparison of edges detected in rendered and real image pairs. However, when the products/parts or lighting conditions change, the parameters of these comparison algorithms have to be adjusted again by subject matter experts. In the current work we introduce a data-driven learning approach to solve this problem. We use the idea of image domain translation to translate the real-world images into rendered like ones, so that the translated and ground-truth images can be compared using simple image comparison measures, thus minimizing involvement of human experts. We use CGAN for the purpose of image domain translation and MSE for the purpose of image comparison. By choosing individual MSE thresholds for different types of parts and for some parts (with extreme aspect ratio) training on image tiles instead of whole image at once, we achieved 100% error detection while mis-classifying only 0.5% correct assembly samples as errors.

References

1. S. Sauer, T. Dunker, and M. Heizmann, "Ein Framework zur Simulation optischer Sensoren," in *20. GMA/ITG-Fachtagung Sensoren und Messsysteme 2019*. Nürnberg: AMA Association for Sensors and Measurement, 2019.
2. S. Sauer and D. Berndt, "Optische Montageprüfung unter Nutzung intelligenter Algorithmen," in *3D-NordOst 2018*, Berlin, 2018, pp. 35–42.
3. M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
4. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
5. P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proceedings CVPR 2017*, vol. 2017-Janua, pp. 5967–5976, 2017.
6. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science*, vol. 9351, pp. 234–241, 2015.
7. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
8. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
9. F. Chollet, *Deep Learning with Python*, 1st ed. USA: Manning Publications Co., 2017.

A Step towards Explainable Artificial Neural Networks in Image Processing by Dataset Assessment

Nina Felicitas Heide*, Alexander Albrecht*,
and Michael Heizmann[+]*

*Fraunhofer IOSB, Fraunhofer Center for Machine Learning,
Fraunhoferstraße 1, Karlsruhe, Germany
{*nina.heide, alexander.albrecht*}@*iosb.fraunhofer.de*
[+] Karlsruhe Institute of Technology,
Institute of Industrial Information Technology,
Kaiserstraße 12, Karlsruhe, Germany
michael.heizmann@kit.edu

Abstract We propose a methodology for generalized exploratory data analysis focusing on artificial neural network (ANN) methods. Our method is denoted *IC-ACC* due to the combined assessment of information content (*IC*) and accuracy (*ACC*) and aims at answering a frequently posed question in ANN research: "What is good data?" As the dataset has the primary influence on the development of the model, *IC-ACC* provides a step towards explainable ANN methods in the pre-modeling stage by a better insight in the dataset. With this insight, detrimental data can be eliminated before a negative influence on the ANN performance occurs. *IC-ACC* constitutes a guideline to generate efficient and accurate data for a specific, data-driven ANN method. Moreover, we show that training an ANN for the semantic segmentation of 3D data from unstructured environments with *IC-ACC*-assessed and -customized training data contributes to a more efficient training. The *IC-ACC* method is demonstrated on application examples for the visual perception of robotic platforms.

Keywords Artificial neural networks, image processing, premodeling explainability, robot vision systems

1 Introduction

In the development of classic methods – without the use of artificial intelligence (AI) – the scientist defines the behavior of a method by domain knowledge. In contrast to this, AI methods can be separated into data-driven and model-driven methods [1]. Initial design considerations in ANNs are specified with expert knowledge. Apart from this, the input data constitutes the major impact on the performance of a data-driven ANN approach [2]. To develop powerful AI methods, it is advisable to examine the input data in the pre-modeling stage.

We classify the data depending on the target application of the ANN. The 2D imaging domain can be divided into segmentation, depth estimation, object detection and tracking, and classification. 3D imaging splits into segmentation, object detection and tracking, shape classification, and registration [3,4].

In explainable AI research, the stages of explainability are subdivided into pre-modeling explainability, explainable modeling, and post-modeling explainability [2]. Pre-modeling explainability includes exploratory data analysis, dataset description standardization, dataset summarization, and explainable feature engineering. So far, most methods for exploratory data analysis examine and summarize the main characteristics and focuses on statistic parameters such as the Google Facets toolkit which maps the characteristics into numeric and categorical features.

Dataset assessment methods can be subdivided into adversarial testing methods, testing methods based on model and data coverage, and testing based on metrics [5]. In coverage testing, a high quality of a dataset is derived form a high percentage of activated neurons [5]. Testing based on metrics mainly focuses on the prediction accuracy of the ANN method and dismisses the in-depth analysis of the underlying dataset. Most research focuses on only one target application such as classification in [5] and [6]. [5] proposes an in-depth method to test the coverage of deep neural network models by examining the dataset quality with statistical measures such as centroid positioning. [6] detects class structure ambiguities in classification and proposes a reorganization strategy in case of decreasing

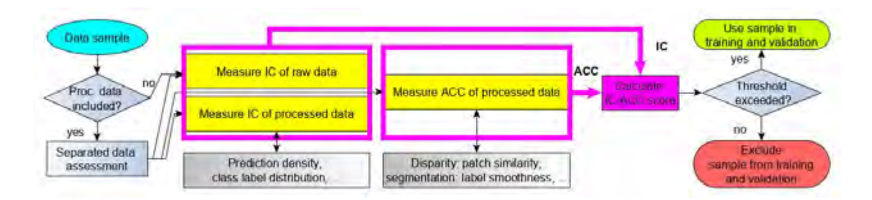


Figure 2.1: IC-ACC method: separated assessment of raw and processed data, the *IC-ACC* score decides whether to include the sample in the final dataset.

accuracy. Usually, the processed reference data is assumed as sufficiently accurate ground truth without verification.

Compared to the extensive research on ANN methods, data analysis for ANN methods is greatly underrepresented. With *IC-ACC*, we contribute a generalized, step-by-step approach in exploratory data analysis for pre-modeling explainability. We focus on the amount and diversity of the information inside the data which is required for proper ANN training as well as on the accuracy of its reference data for supervised learning approaches. Hence, we target the step prior to dataset assessment as proposed in [5] which analyzes biases or correlation among the variables. Only classical methods are considered in *IC-ACC* as AI-based data analysis would require additional assessment.

2 Exploratory Dataset Analysis with *IC-ACC*

Contrasting most works on exploratory data analysis, *IC-ACC* focuses on a generalized assessment of training data for ANN methods in the domain of image processing. The workflow of the proposed *IC-ACC* method is illustrated in Fig. 2.1. *IC-ACC* does not only provide a statistical measure for the diversity of the information such as the number of categories. But, we combine this with information content measures for images and point clouds, which can optimize the ANN performance, as well as with a first step towards an accuracy analysis of datasets.

2D and 3D data divides into raw and processed data: is the output of a sensor system after applying the intrinsic calibration and

is assumed to be error-free, processed data designates the reference data obtained in processing of the raw data which is usually utilized as training data. As the faultlessness of in data processing cannot be guaranteed, processed data can be subject to errors. With supervised training, an ANN learns to interpret raw data and requires processed data as a reference for the training loss. For unsupervised training, raw data is sufficient.

2.1 Information Content *(IC)* of Raw and Processed Data

To assess the information content of a message, Shannon [7] proposed the entropy measure $H = -\big(\sum_{N_I} p(i) \cdot \log_2(p(i))\big)$, with $i \in I$ a single symbol of all available symbols I, $p(i)$ the probability of the symbol to occur in the message, and N_I for $\#I$. In *IC-ACC*, the Shannon entropy is transferred on 2D and 3D data to assess the respective *IC*.

The *IC* of raw 2D data (IC_{r2D}) is contained in the intensity values of the captured spectral channels inside a pixel structure. The *IC* of a defined 2D pixel grid can be measured by its Shannon entropy. In 8-bit images, I contains all possible intensities $I \in \{0, 1, ..., 255\}$.

Raw 3D data provides geometric information in 3D space with the position of each measurement point. Following [8] and [9], we regard point density and geometric structure as most conclusive criteria for the *IC* (IC_{r3D}). For point density, the density related to the distance from the sensor origin is chosen as the most promising representation [9]. 3D data is transformed from the Cartesian coordinates into homogenized coordinates ϕ, r, and z: $\phi = \arcsin\left(y/\sqrt{x^2+y^2}\right)$, $r = \sqrt{x^2+y^2}$, and $z = z$. This yields a more uniform point distribution for active, rotating sensors [9]. A normalization in r and the binning of the homogenized points by their values of ϕ illustrates the point distribution, and thus the density. If notably different areas are included in each cloud, it is advisable to set a high number of bins. As a higher number of points naturally denotes a higher *IC*, also the total number of points as well as the number of points inside each previously defined bin can be applied to compare samples. We calculate the empirical mean to represent the relative distribution of the density values: $\mu = {}^1\!/_N \sum_{i=1}^{N} x_i$, with N the number of bins, and x_i the relative density inside bin i. A uniform point distribution –

and thus a high μ – illustrates a proper representation of all cloud sectors and thus a high *IC*.

The structure of the point cloud can be described with the surface variation $s = \lambda_3/(\lambda_1+\lambda_2+\lambda_3)$, with λ the eigenvalues when decomposing the covariance matrix of a point set. s is calculated for each point and indicates the structured or unstructured character of a point cloud. To combine the values s of a point set, we calculate the Gaussian mean of s, denoted $\bar{s}$. In general, structured environments contain controlled, clearly separable topological objects as well as a high number of smooth surfaces. Unstructured environments are dominated by natural elements such as grassland, trees, bushes, or rocks [9, 10]. Hence, a higher $\bar{s}$ indicates less structured elements. The future application environment defines if a high or a low *IC* measure is achieved: for more complex, unstructured environments, a high $\bar{s}$ indicates a high *IC*, while in structured environments a high *IC* is synonymous to a clear structure and thus a low $\bar{s}$. This selection is justified in the subsequent proof of concept.

The *IC* of processed data depends on the prediction density and diversity of the information added during the processing step in relation to the number of 2D pixels or 3D points. For 2D data (IC_{p2D}), the prediction density is related to the number of pixels. An example for 2D prediction density is provided in stereo depth estimation: a high prediction density indicates a high percentage of valid depth estimates and thus a decent quality of the reference data [11]. For 3D data (IC_{p3D}), the number of points inside a point cloud is used accordingly. The diversity of the information can be measured using the Shannon entropy. Here, I contains all possible values of the added information such as class labels in segmentation tasks. The relative frequency of these values determines $p(i)$.

2.2 Accuracy *(ACC)* of Processed Data

ACC provides a measure of confidence and error characteristics for the data used as reference in supervised training. To overcome the common lack of a verified, error-free ground truth to compare against, we assess *ACC* with indirect measures. In contrast to *IC*, the evaluation of *ACC* has to be adapted to the type of the processed information to some extent. Two groups can be distinguished:

data that is used to train an ANN for similarity matching (ACC_{2Ds}, ACC_{3Ds}) and data for interpretation (ACC_{2Di},ACC_{3Di}). Similarity includes depth estimation in 2D and the registration in 3D, whereas segmentation, object detection and tracking, and classification aim at the interpretation of imaging data.

In ACC_{2Ds} and ACC_{3Ds}, source and target to be matched have to be examined for similarity. The target remains in its original representation, the source is transformed by applying the reference data to be assessed. Following [11], the similarity of 2D samples for depth estimation is measured using the structural similarity index measure (SSIM) and the normalized root mean squared error (NRMSE). For 3D registration, the processed data typically consists of transformations. A high similarity between source and target cloud, after applying the reference transformation to the source, indicates a high *ACC*. Hence, difference measures such as L_1 norm, L_2 norm, or NRMSE are applicable. NRMSE generates a scale-invariant difference measure, which can be problematic in case of an undetected, different scaling of the input data. Hence, the L_2 norm is applied to assess the similarity [9].

For ANN methods in interpretation (ACC_{tDi}) the processed data typically consists of labeling information. One obvious strategy is to check a small number of random samples manually and to deduce a qualitative statement. This is a time-consuming, but often a straightforward strategy for experts. Currently, human annotators generate labeling data with the assistance of labeling tools and errors tend to occur in border regions or transitions between objects. As objects are rarely represented by a small number of points or pixels, a high number of different labels in a small area or space can indicate noisy and inaccurate data. As a first step towards a verifiable and quantitative *ACC* measure, pixel- and point-wise labels can be examined for smoothness, and thus for the existence of outliers. To identify outliers automatically, a nearest neighbor search can be applied similar to [8], requiring a minimum number of neighbors with identical labels. Also, a qualitative visual assessment of label smoothness is possible. A scoring from 0 to 10 allows a detailed rating for experts with domain knowledge [12]. Here, 10 stands for the highest *ACC* possible.

Table 1: Elements of exploratory data analysis with *IC-ACC* with proposed measures.

IC-ACC element	Measure
IC_{r2D}: raw 2D	Shannon entropy H
IC_{r3D}: raw 3D	Surface variation $\bar{s}$, relative density μ
IC_{p2D}: processed 2D	Prediction density (pixels), diversity H
IC_{p3D}: processed 3D	Prediction density (points), diversity H
ACC_{2Ds}: similarity	NRMSE, SSIM
ACC_{2Di}: interpretation	Qualitative visual assessment, label smoothness
ACC_{3Ds}: similarity	L_2 norm (MSE)
ACC_{3Di}: interpretation	Qualitative visual assessment, label smoothness

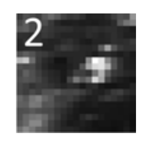
2.3 Deriving the *IC-ACC* Score

To derive a holistic *IC-ACC* score, each *IC* and *ACC* measure is normalized individually. Tab. 1 provides an overview of all *IC-ACC* elements. The *IC-ACC* score is calculated using $IC\text{-}ACC = 1/3 \cdot (IC_{\mathrm{rtD}} + IC_{\mathrm{ptD}} + ACC_{\mathrm{ptD}})$, with $t \in 2, 3$. If more than one measure is included in an *IC-ACC* element, the average of both measures is considered. For normalization, the respective values are mapped to $[0, 1]$ using the maximum value of the respective measure. Depending on data availability, we recommend to distinguish weak, medium, and strong data inclusion thresholds for the *IC-ACC* score. In reference to $\mathcal{N}_{0,1}$, we include samples achieving more than 68.27 % of the possible maximum *IC-ACC* score of 1.0, hence it is *IC-ACC* score > 0.6827 for a weak threshold. The medium threshold is set to 0.8664 ($\mu \pm 1.5\sigma$), the strong threshold is 0.9545 ($\mu \pm 2\sigma$). Regarding one dataset in-depth, bad samples can be detected by applying the threshold on all elements of the dataset. To compare different datasets, the *IC-ACC* scores, prior to and after applying the inclusion requirements, can be compared.

2.4 Proof of Concept: *IC-ACC* for 2D and 3D Data

For the IC_{r2D}, the Shannon entropy H of an image is calculated. Fig. 2.2 shows the IC_{r2D} and ACC_{2Ds} assessment of image patches used to train a CNN for stereo matching on the KITTI 2012 dataset







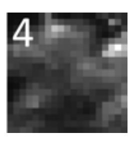


Figure 2.2: IC_{r2D} and ACC_{2Ds} for 19×19 pixel patches to train a depth estimation CNN: patch 1 has a low IC with $H_1 = 2.5$, whereas $H_2 = 6.24$ and $H_3 = 5.75$ indicate a high IC; for pair 3-4 SSIM = -0.04 is sufficient, but NRMSE = 1.15 is too high for similarity [11] and the reference disparity is rated as inaccurate.

[13] using the reference disparity for ACC_{2Ds} as proposed in [11]. The prediction density concept for IC_{p2D} is illustrated on disparity maps in [11].

We demonstrate the *IC* analysis for raw 3D data on sequence (seq.) 00–10 of the SemanticKITTI dataset [4] using the raw 3D data of the KITTI Vision Odometry Benchmark [13] captured in urban and sub-urban areas. As terrain in urban and suburban areas mostly includes cultivated and rather structured terrain, only two of the 28 classes predominantly represent unstructured elements: vegetation and trunk.

In case of clearly separable sectors in point clouds, a subdivision into sectors improves the in-depth *IC* analysis. The clouds are divided into four sectors of 90° which are axisymmetric to the axes of the Velodyne LiDAR frame: front, right, back, and left. For seq. 02–04, $\bar{s}$ of the left and right sectors is notably higher than $\bar{s}$ of front and back: $\bar{s}_{\mathrm{left,02–04}} = 0.0460$, $\bar{s}_{\mathrm{right,02–04}} = 0.0649$, $\bar{s}_{\mathrm{front,02–04}} = 0.0211$, and $\bar{s}_{\mathrm{back,02–04}} = 0.0228$. Targeting on unstructured environments, this shows a higher *IC* for the left and right sectors and justifies the separation into structured and unstructured sectors for SemanticKITTI. Fig. 2.3 shows the point-wise estimates of s in scene 245 of seq. 04. Tab. 2 shows the $\bar{s}$ and the class distribution for seq. 01 and 09 with the highest and seq. 06 with the lowest $\bar{s}$ in seq. 00-10 of SemanticKITTI. Seq. 06 consequently has the lowest *IC* for unstructured target environments. As in seq. 06 only 9.77 % of the labels represent unstructured classes, compared to 23.91 % and 29.96 % in the seq. 01 and 09, this justifies the surface variation metric as a measure for the structured or unstructured character of a point cloud.

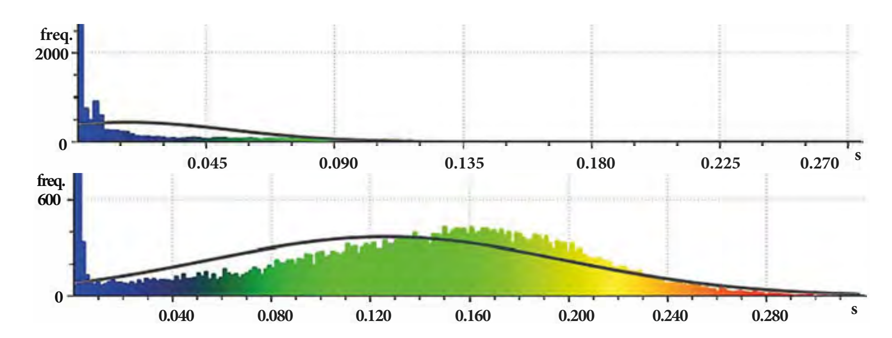


Figure 2.3: IC_{r3D}: histogram of surface variation s in the front sector (above) and right sector (below) of scene 245, seq. 04. The normal estimation radius is 0.40 m [8]. The low $\bar{s} = 0.0181$ of the front sector highlights its structured nature, while the high $\bar{s} = 0.1277$ of the right sector shows its unstructured character. Frequency for cut-off bin 0 is 13515 (front) and 3783 (right).

To demonstrate *IC-ACC* in the comparison of two 3D clouds, we select scene 245 of seq. 04, denoted as (245,04), with a medium $\bar{s}$ and scene 778 of seq. 09 (779,09) with a high $\bar{s}$. Tab. 3 illustrates the *IC-ACC* results. The point density in IC_{r3D} is calculated using $N = 12$ bins, mapping 30° in one bin. ACC_{3Di} is demonstrated in qualitative manner. The renowned SemanticKITTI dataset comes with a high labeling accuracy and label smoothness which can both be verified manually. For IC_{p3D}, $N_I = 28$ is set in H with 28 classes in SemanticKITTI. In 245,04, the most frequent classes are road and vegetation with 35.46 % and 20.36 %. In 778,09 it is 26.89 % vegetation and 17.87 % building. The normalization values are derived from the maximums such as $\max(\bar{s}) = \bar{s}_{778,09}$ for s. We get $IC\text{-}ACC_{245,04} = 1/3 \cdot \big((0.574 + 1.0)/2 + (0.828 + 1)/2 + 1.0\big) = 0.90$ and $IC\text{-}ACC_{778,09} = 1.0$. Applying a weak or medium threshold, both samples exceed the requirement with 86.64 % $<$ 90.0 %. For a strong threshold only 778,09 would be included in the final dataset.

2.5 Semantic Segmentation Efficiency with *IC-ACC*

The benefits of data analysis in training an ANN for the semantic segmentation of 3D point clouds is shown with SqueezeSeg [14, 15]. We follow the implementation of [15], but train with the full and

Table 2: $\bar{s}$ and class distribution of points (in %) for lowest and highest $\bar{s}$ in seq. 00-10.

$\bar{s}$	Vegetation	Trunk	Mainly unstructured	Terrain
$\bar{s}_{01} = 0.051$	23.87	0.04	23.91	13.83
$\bar{s}_{06} = 0.027$	9.31	0.46	9.77	26.10
$s_{09} = 0.051$	29.29	0.67	29.96	8.88

with a reduced version (seq. 02–04 training, seq. 08 validation) of SemanticKITTI [4]. Intersection over Union (IoU) is used to measure the segmentation performance. Training is conducted for 150 epochs. Representative classes are grouped into structured (car, road, parking, sidewalk, building, fence, pole, traffic sign) and unstructured, nature classes (vegetation, terrain, trunk). The full dataset achieves a mean IoU of $\overline{\text{IoU}} = 0.173$ in training and 0.210 in validation, the reduced dataset reaches 0.166 and 0.175. For the nature classes, it is $\overline{\text{IoU}} = 0.335$ on the reduced dataset, the structured classes reach $\overline{\text{IoU}} = 0.266$. This is remarkable as the nature classes are attributed to less than 30 % of the points present in the training data. It underlines the statement that unstructured data has a higher *IC* making it favorable in training and inference due to an increased unambiguousness and a higher *IC*.

To rate the training efficiency, the customized segmentation efficiency metric $\eta_{\text{IoU}} = \text{IoU}/(D_T \cdot N_T)$ is proposed. The clouds are subdivided into structured (front, back) and unstructured (left, right)

Table 3: *IC*-*ACC* assessment of scene 245, seq. 04 and scene 778, seq. 09.

	Raw measures		Normalization	
Measure	245, 04	778, 09	245, 04	778, 09
IC_{r3D}: $\bar{s}$	0.027	0.047	$\frac{0.027}{0.047} = 0.574$	$\frac{0.047}{0.047} = 1.0$
IC_{r3D}: μ	0.083	0.083	$\frac{0.083}{0.083} = 1.0$	$\frac{0.083}{0.083} = 1.0$
IC_{p3D}: H	2.405	2.903	$\frac{2.405}{2.903} = 0.828$	$\frac{2.903}{2.903} = 1.0$
IC_{p3D}: pred. density	100 %	100 %	$\frac{100\%}{100\%} = 1.0$	$\frac{100\%}{100\%} = 1.0$
ACC_{3Di}: qual.	10, 10	10, 10	$\frac{10}{10} = 1.0$	$\frac{10}{10} = 1.0$
IC-*ACC* score	–	–	0.90	1.0

sectors. D_T represents the amount of data inside each tensor and is calculated for each scene with $D_T = hwc$. It is $h = 64$ and $w = 512$, height and width of the spherical projection, and $c = 5$ the number of features per point. The separation into sectors yields four projections with $4 \cdot D_T$ data points per cloud. N_T denotes the number of randomly selected scenes from the reduced dataset that are used in training. Hence, η_{IoU} measures the IoU in relation to the amount of training data. We test $N_T \in \{350, 700, 1050, 1750, 2800\}$. Overfitting is evaluated using the front and right sectors and can be prevented with $N_T \geq 1050$. For reference, it is $\eta_{\text{IoU}} = 3.1 \cdot 10^{-8}$ using all four sectors with N_T = 2800. Also with $N_T = 2800$, it is $\eta_{\text{IoU}} = 6.0 \cdot 10^{-8}$ using the right sectors only, $\eta_{\text{IoU}} = 3.9 \cdot 10^{-8}$ combining the two unstructured sectors, and $\eta_{\text{IoU}} = 5.2 \cdot 10^{-8}$ using the unstructured left and the structured front sectors. This shows that a notably higher η_{IoU} is achieved with a similar amount of training data, but with different structure. For seq. 02–04 of SemanticKITTI, the $\overline{\text{IoU}}$ can be raised by more than 30 % combining data with different surface variations instead of data with a similar structure. Hence, the composition of *IC*-efficient datasets can improve the performance of ANN methods or reduce the amount of labeled training data required to achieve comparable results.

2.6 Guidelines for Data Generation

Naturally, guidelines for future data generation can be derived from the proposed *IC-ACC* method. We recommend to ensure that the captured data achieves a high *IC* and a high *ACC*. With this, the central point in the generation of data is fulfilled: it does neither contain too similar or too little, nor erroneous information. To ensure this, test samples can be assessed prior to capturing the final dataset. For 3D data, the target application of the ANN method defines the desired surface variation as previously stated. Capturing 3D for applications in unstructured environments such as in off-road robotics, a high $\bar{s}$ is required, whereas a dataset for indoor scenes rather requires a low $\bar{s}$ measure. Furthermore, we recommend to apply the *ACC* measures on the test data samples to verify a high *ACC* for the full, subsequently generated dataset.

3 Conclusion and Future Work

We present *IC-ACC*, a generalized methodology for exploratory data analysis for ANN methods in image processing. The *IC* examination can be applied to filter detrimental data and facilitates the composition of efficient datasets. The proposed *ACC* measures present a first step towards confidence and error assessment for supervised learning data. We demonstrate *IC-ACC* on ANN methods for robotic perception. Applying *IC-ACC* in the semantic segmentation of 3D data from unstructured environments shows an increased performance when using properly assessed and customized training data. Furthermore, *IC-ACC* presents a guideline for an efficient and less error-prone data generation. As data-driven AI methods can learn erroneous behaviors from erroneous training data, the analysis of the input data is an important step towards reliable and explainable AI methods.

Future works include evaluating *IC-ACC*-efficient data for ANN methods in image processing as well as extending *IC-ACC* to other domains.

Acknowledgement: The described research has been conducted within the competence center "ROBDEKON – Robotic Systems for Decontamination in Hazardous Environments", which is funded by the Federal Ministry of Education and Research (BMBF) within the scope of the German Federal Government's "Research for Civil Security" program.

References

1. R. Ashri, “Building AI software: Data-driven vs model-driven AI and why we need an AI-specific software development paradigm,” *URL: https://hackernoon. com/building-ai-softwaredata-driven-vs-model-driven-ai-and-why-we-need-an-aispecific-software-640f74aaf78f*, 2018.
2. B. Khaleghi, “The How of Explainable AI: Pre-modeling Explainability,” *URL: https://towardsdatascience.com/the-how-of-explainable-ai-pre-modelling-explainability-699150495fe4*, 2019.
3. Y. Guo et al., “Deep Learning for 3D Point Clouds: A Survey,” *arXiv preprint arXiv:1912.12033*, 2019.

4. J. Behley et al., "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," *ICCV*, 2019.
5. S. Mani et al., "Coverage Testing of Deep Learning Models using Dataset Characterization," *arXiv preprint arXiv:1911.07309*, 2019.
6. J.-D. Wang and H.-C. Liu, "An approach to evaluate the fitness of one class structure via dynamic centroids," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13 764–13 772, 2011.
7. C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.
8. N. Heide, T. Emter, and J. Petereit, "Calibration of multiple 3D lidar sensors to a common vehicle frame," *ISR*, 2018.
9. N. F. Heide et al., "UCSR: Registration and Fusion of Cross-Source 2D and 3D Sensor Data in Unstructured Environments," *Fusion*, 2020.
10. J. Petereit et al., "ROBDEKON: Robotic Systems for Decontamination in Hazardous Environments," *SSRR*, 2019.
11. N. F. Heide, S. Gamer, and M. Heizmann, "UEM-CNN: Enhanced Stereo Matching for Unstructured Environments with Dataset Filtering and Novel Error Metrics," *ISR*, 2020, in press.
12. N. F. Heide, A. Albrecht, and M. Heizmann, "SET: Stereo Evaluation Toolbox for Combined Performance Assessment of Camera Systems, 3D Reconstruction and Visual SLAM," *ICICSP*, 2019.
13. A. Geiger et al., "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
14. B. Wu et al., "SqueezeSeg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3D lidar point cloud," *ICRA*, 2018.
15. A. Milioto et al., "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation," *IROS*, 2019.

Extraction of surface image features for wear detection on ball screw drive spindles

Tobias Schlagenhauf, Max Heinzler, and Jürgen Fleischer

Karlsruhe Institute of Technology (KIT),
wbk Institute of Production Science,
Kaiserstraße 12, 76131 Karlsruhe

Abstract Failures of production machines are often caused by wear and the resulting failure of components. Therefore, condition-based monitoring of machines and their components is becoming an increasingly important factor in industry. Due to the simple conversion of the motion of electric rotary drives into precision feed motion, the ball screw is an inherent element of many production machines. Thus, a failure of the ball screw often leads to costly production stops. This paper shows the determination and extraction of wear-describing image features, allowing an image-based condition monitoring of ball screws using hyperparameter-optimized machine learning classifiers. The features to train the algorithms are derived and extracted based on the deep domain knowledge of ball screw drive failures in combination with further developed state of the art feature extraction algorithms.

Keywords Ball screw drive, image features, artificial intelligence, machine learning, pattern recognition

1 Introduction

The changes in the economic and technological environment force companies to constantly review their positioning in relation to their competitors and to search for innovations and competitive advantages. A decisive competitive advantage is an effective and efficient production. To avoid downtimes, the machines must be in perfect

condition [1]. A common goal in industry is to extend the lifecycle of production systems by early identification of defects and damages to machine parts. Such approaches are called Preventive and Predictive Maintenance. In the past, repairs and maintenance have been executed after machine breakdowns or a fixed period. Today, companies try to schedule repair and maintenance activities depending on the estimation of a machine's condition [2]. Knowing the right time to replace a machine's component is a desirable situation from a technical and economic perspective. If the components are changed too late, there will be a risk of damaging other machine elements or manufacturing faulty products, which in both cases will result in financial disadvantages. If the component is replaced too early, a certain part of its lifecycle will remain unused and unnecessarily premature financial expenses are incurred. A majority of machines used in the manufacturing process includes rotating components. Ball screws are the most frequently used design elements in today's machine tools for converting the motion of rotary electric drives into precision feed motions. Therefore, the functionality of a machine depends on the ball screw and a failure can lead to a costly production stop [3]. Typical reasons for such defects are abrasion by foreign particles, adhesion due to cold welding and surface disruption. Surface disruption occurs during application and results in pittings. Pittings are a common reason for ball screw failure, so the aim of this study is to detect this wear automatically.

In literature, the condition of a ball screw is often analyzed through vibration. As described in [4] and [5], most mechanical systems generate vibration signals that provide information about the state of a system. The more relevant approaches for this work are the image-based methods of defect analysis. Approaches from other metallic surfaces than spindles are also considered to investigate further possible solutions. A frequently applied method is the use of deep learning algorithms. In most cases, these approaches are based on a convolutional neural network (CNN). As described in [6] and [7], this technique has already been successfully applied to ball screws to detect pitting using a CNN. The classification accuracy of [6] is just over 90% and that of [7] is even higher at 99%. [8] adopt the deep learning approach to analyze image data for the identification of rail surface defects. The algorithm also classifies

among various defect conditions (normal, weld, light squat, moderate squat, heavy squat and joint). The results of the approach prove the efficiency with a classification accuracy of almost 92%. Another CNN-based approach is published by [9]. This method has also been developed to monitor defects in the rail system.

Deep learning is an effective approach with the disadvantage of lacking traceability of the decision making process. The extraction of image features using domain knowledge and subsequent classification increases the transparency of decision making.

2 Ball screw drive image features

The first subject is the preparation of the image data set. It is important to create a comprehensive data set in order to be able to record as much optical characteristics as possible. Since pittings occur in the thread raceway, it is the region of interest (ROI). The thread ridges have a strong optical characteristic and are therefore not included in the data set. As a result, only the thread raceways are extracted. Also, a suitable image size must be selected in order to be able to analyze as much ROI information as possible. This leads to an image resolution for the single images of 128x128 pixels and the data set size is 1000 images per class (pitting and no pitting).

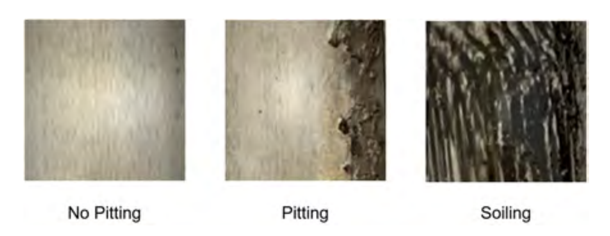


Figure 2.1: Sample images from the data set

A main task of the paper is to extract features from the wear patterns of the ball screw with image filter methods based on domain knowledge. The challenge is the automatic detection of wear on the spindle (Pittings) despite oil residues on the ball screw spindle. Since soiling has a strong optical characteristic, it must be also con-

sidered in detail and thus represents the third class in the analysis besides pitting and no pitting (see Table 1). For classification, only the classes pitting and no pitting are used. The elaborated characteristics are assigned to the image feature categories color, shape and texture. Pittings occur in the raceway, so only this area of the spindle is considered in the analysis. Figure 2.1 shows a sample images from the data set without pitting, an image with pitting and an image with soiling.

Since color features are invariant to scaling, translation and rotation, they have a major impact on image analysis. The spindle surface, soiling and pittings are exclusively brown and grey shades, which is why the share of red, green and blue (RGB) is almost equal. A surface of a spindle raceway without pitting and without soiling is characterized by having almost exclusively bright brown and grey shades. A significant difference can be detected in the images with pitting. Because of the dark pittings, the image is not composed exclusively of bright colors like the image without them. The soiling on the spindle is usually oil residues, therefore, it is typically black or grey. In the case of heavy soiling, oil residues can cover the entire raceway, so that the color composition consists mainly of very dark shades. Due to the nearly identical color of pitting and soiling, an almost similar histogram can be determined for an image without pitting, but soiling. For this reason, texture and shape features are determined in addition to the color features.

A spindle without pitting usually has a uniform raceway structure. The surface has hardly any or no contrast differences. The balls of the ball screw result in a slight vertical structure in the running direction. To a great extent, this structure is also evident in the image with pittings. Pittings have an uneven structure with a strong contrast difference to the rest of the raceway. The texture in the area of the pitting appears partially plateau-like, interspersed with dark spots. Pittings are not uniform but have varying degrees of protrusion into the rest of the material without pittings. Since soiling is random, there are often many local and global contrast differences and the surface can have a random and/or uniform texture. However, soiling often follows the vertical running direction of the screw drive as shown in the sample image (see Figure 2.1). Soiling is the major unknown factor in the texture analysis due to its random occurrence.

Considering the raceway of a spindle without pitting, no noticeable shapes can be identified. With pitting and soiling, it is not the shape itself that is decisive, as soiling can have similar contours to pittings, but the location of the shape is an indication of the respective class. Pittings always occur in the flanks of the spindle raceways, while soiling usually spreads over the entire surface or occurs in the middle area of the raceway. Due to the knowledge of the occurrence of soiling and pittings, the shape features are ideally suited as spatial features.

Table 1: Differences No Pitting/Pitting/Soiling based on domain knowledge

No Pitting	Pitting	Soiling	Feature Category
Light brown, grey	Dark brown shades	Black, grey	Color
Few color shades	Many different color shades	Few color shades	Color
No colored line	Partly colored line next to pitting	No colored line	Color
Regular surface	Irregular surface	Uniformity of the surface depends on the degree of soiling	Texture
Few contrast differences	Many global/local significant differences in contrast	Significant differences in contrast; quantity depends on the degree of soiling	Texture
Even texture	Random texture	Texture runs in the direction of the thread balls	Texture
No plateau-like texture	Plateau-like/"Map"	No plateau-like texture	Texture
Hardly any edges	Many edges	(Mostly) many edges	Shape
-	Occurs on the flank	Occurs randomly; usually spread over the entire surface	Spatial

3 Feature extraction

Table 1 is fundamental for the following development of the extraction methods. The methods are intended to extract a variety of characteristics from this table.

3.1 Color Features

Each image contains 49152 color values (128x128 pixels × RGB values). The extraction approach for the color features focuses on simplifying and clustering the data. To cluster the image colors an own approach based on [10] is develpoed. The K-Means algorithm is applied and the pixels are assigned to a defined number of clusters. The color of a cluster is defined by its centroid. With this method, the color properties are displayed in a more compact form but to capture a wide color spectrum of an image, a high number of clusters must be defined. Therefore, the number of clusters is set to 20. To describe the relationship between color and distribution of the clusters the own approach "Clustered Color Share (CCS)" is applied. Since the RGB values for each cluster are almost identical due to the predominantly grey or brown colors of the data set images, the RGB mean value of the centroid is calculated. Afterwards, the averaged RGB value of the centroid is multiplied with the cluster's share. This way the color as well as the share can be represented in one feature. The progression of the twenty features and the feature values can be used to identify the composition of the color in an image. The mean value, median value, maximum value, minimum value and standard deviation of the RGB mean values are additionally included as features in the feature vector.

3.2 Texture Features

As a result of the high complexity of the texture, two approaches from literature are applied and examined. The first approach combines the grey-level co-occurrence matrix and the Haralick features [11] to compute a global representation of the texture. Initially, the four grey-level co-occurrence matrices are calculated and afterwards,

the Haralick features are determined. The feature vector is the average of the result vectors of the individual matrices. For this purpose, a matrix with the 13 Haralick features is created and the feature vector for the images is calculated.

The second approach uses the Local Binary Patterns (LBP) [12] to compute a local representation of texture. The first step is to create the LBP matrix for the image. For this purpose, a 3x3 pixels neighbourhood is chosen for each pixel. Following the calculation of the matrix, the first and last columns as well as rows are truncated because no calculation is possible for these pixels as they have no 3x3 neighbourhood. Afterwards, the frequency of the individual LBP patterns can be determined and saved as feature vector. As additional features, the statistical properties mean, median, minimum and maximum of the feature vector are appended.

3.3 Shape/Spatial Features

The extraction of the shape features is based on the SIFT algorithm published by [13]. The SIFT method enables the search for features that are invariant to rotation, translation, scaling, changes in light conditions and partially affine distortion [13]. However, the resulting derivation of the shape/spatial feature and the extraction of the feature vector is the own approach "KeyPoints Per Sub Region (KPPSR)". Since pittings occur on the flanks of a spindle, the shape features are ideally suited to describe the spatial variable of pittings. The position and number of KeyPoints extracted from the SIFT algorithm are used to describe this characteristics. The number and location of the KeyPoints can provide information about the structure and shape of the object (see Figure 3.1).

As exemplarly shown in Figure 3.1 the distribution of the KeyPoints over sub regions is an important characteristic to distinguish between pitting and no pitting images. The shape of the oil residues leads to a strong accumulation of KeyPoints over the entire image. To analyze the differences between the inner and the outer regions of the image, the image is divided into 4x4 sub regions and the SIFT algorithm is applied. After the segmentation of the image into sub regions, the feature vector can be determined. Thereby, the sub regions form a matrix. The regions are numbered (from 0 to 3) in x

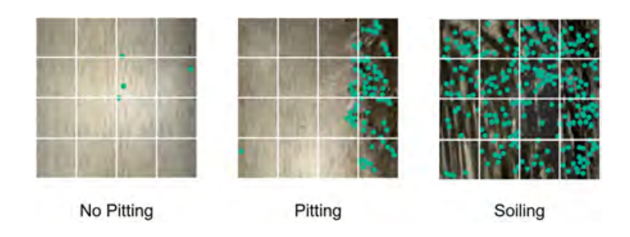


Figure 3.1: Scheme "KeyPoints per Sub Region"

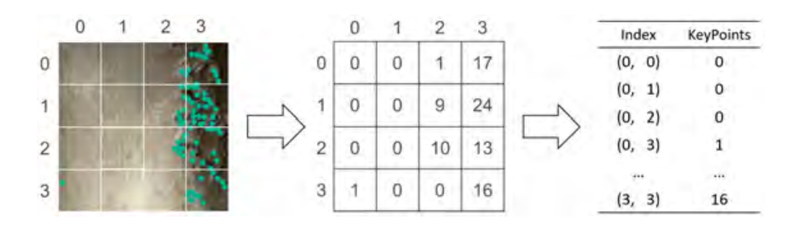


Figure 3.2: Example "KeyPoints per Sub Region"

and y direction resulting in each region having a unique index value (see Figure 3.2).

4 Results

Overall, 28 color features, 274 texture features (261 LBP, 13 Haralick) and 16 shape features are extracted. To verify the performance of the extracted features, three-layer neural networks are applied to the individual methods. Since the optimal number of neurons per layer, the optimal activation function and the optimal solver can be different for each feature, 100 randomly generated combinations are applied to the features. Due to their stochastic nature, neural networks behave slightly different for each training. Therefore, each hyperparameter combination is applied five times. This means that a total of 500 neural networks are applied to the individual methods. All further tests are executed with the same split data sets. In total, the 2000 extracted feature data sets are randomly divided into 1600 training data sets (791 No Pitting and 809 Pitting) and 400 test data

sets (209 No Pitting and 191 Pitting). Table 2 shows the average fitness of the 500 neural networks, the average fitness of the ten best neural networks and the fitness of the best model.

Table 2: Results Methods

Method	Average Fit	AvgTop10	Best Fit
KPPSR	0,848	0,906	0,915
CCS	0,836	0,893	0,908
Haralick	0,847	0,939	0,950
LBP	0,836	0,929	0,935

All methods alone achieve classification accuracies of over 90%. The best results are achieved with texture features, followed by spatial features and color features. In the next step neural networks are applied to the combination of all features. Furthermore, the own approaches are replaced by existing approaches to compare the performance. For the analysis of the color features, a color histogram is selected and for the analysis of the KeyPoints the total number of KeyPoints in the image is used as feauture. Table 3 shows the average fitness of the 500 neural networks, the average fitness of the ten best neural networks and the fitness of the best model. The best result is achieved with KPPSR without CSS (but with color histogram) at a fitness of 98.8%.

Table 3: Results

KPPSR	CCS	Average Fit	AvgTop10	Best Fit
no	no	0,904	0,978	0,983
yes	no	0,921	0,986	0,988
no	yes	0,897	0,972	0,978
yes	yes	0,914	0,980	0,983

The number of possible combinations of parameter options for classification models are potentially infinite. For such optimization problems, exact methods like exhaustive searches become inefficient and heuristic methods become more suitable. One of these heuristic approaches is the Genetic Algorithm (GA) [14]. Therefore, in the

next step a genetic algorithm is applied to find the optimal hyperparameters for a neural network, which is applied to the extracted features (CCS, LBP, Haralick, KPPSR). The analogy to natural evolution enables genetic algorithms to overcome many of the hurdles that traditional search and optimization algorithms encounter. Especially when problems with a large number of parameters and complex mathematical representations are involved [14] [15]. Using the GA to optimize the hyperparameters of the neural net, classification accuracies of 98.8% can be achieved, which corresponds to the best fitness in Table 3.

In the last attempt, the image features of a spindle area (see Figure 4.1) are extracted and classified using the best-fit neural network. The individual images are recorded using the Sliding Window method. The four frames of the upper row are assigned with the label 0 to the class "No Pitting" and the frames of the lower row are assigned with the label 1 to the class "Pitting". All images are assigned to the correct class.

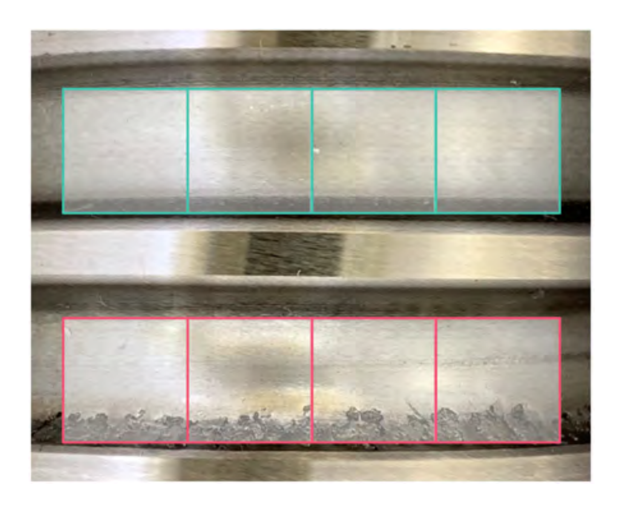


Figure 4.1: Sliding Window

5 Conclusion

Since the ball screw is used in most machines as electromechanical feed drive, the condition of the ball screw is critical for the operation of the machines. Early detection of wear on the spindle, and thus failures, helps to avoid production downtimes and reduce costs. The present approach shows that using a combination of the developed CSS and KPPSR methods together with methods from the literature, features can be extracted to properly classify 98.8% of the spindle surface images for pitting and no pitting. It can therefore be assumed that the selected extraction methods adequately describe the surface of a ball screw. This allows to react to failures at an early stage. Based on the data set, the texture features are the most important features due to the high classification accuracies (see Table 2), followed by the spatial features and color features. The results rely on selected methods and confirm the assumptions through domain knowledge. The hypothesis that texture features are the most important characteristics has to be validated by further experiments to make a general statement.

References

1. A. R. Mohanty, *Machinery Condition Monitoring - Principles and Practices*. Boca Raton, Fla: CRC Press, 2014.
2. D. G. Pascual, *Artificial Intelligence Tools - Decision Support Systems in Condition Monitoring and Diagnosis*. Hoboken: CRC Press, 2015.
3. A. Spohrer, "Steigerung der Ressourceneffizienz und Verfügbarkeit von Kugelgewindetrieben durch adaptive Schmierung," Ph.D. dissertation, Karlsruher Institut für Technologie (KIT), Karlsruhe, 2019.
4. W. G. Lee, J. W. Lee, M. S. Hong, S.-H. Nam, Y. Jeon, and M. G. Lee, "Failure diagnosis system for a ball-screw by using vibration signals," *Shock and Vibration*, 2015.
5. L. Zhang, H. Gao, J. Wen, S. Li, and Q. Liu, "A deep learning-based recognition method for degradation monitoring of ball screw with multi-sensor data fusion," *Microelectronics Reliability*, pp. 215–222, 2017.
6. T. Schlagenhauf, C.-P. Feuring, J. Hillenbrand, and J. Fleischer, "Camera based ball screw spindle defect classification system. System zur

kamerabasierten Defekterkennung auf Kugelgewindetriebspindeln," in *Production at the leading edge of technology. Proceedings of the 9th Congress of the German Academic Association for Production Technology (WGP), September 30th - October 2nd, Hamburg 2019*, J. P. Wulfsberg, W. Hintze, and B.-A. Behrens, Eds. Singapore: Springer Nature, 2019, pp. 503–512.

7. T. Schlagenhauf, P. Ruppelt, and J. Fleischer, "Detektion von frühzeitigen oberflächenzerrüttungen," *wt Werkstattstechnik online*, vol. 110, pp. 501–506, 2020. [Online]. Available: https://e-paper.vdi-fachmedien.de/webreader-v3/index.html#/2657/50
8. S. Faghih-Roohi, S. Hajizadeh, A. Nunez, R. Babuska, and B. d. Schutter, "Deep convolutional neural networks for detection of rail surface defects," *International Joint Conference 2016*, p. 2584–2589, 2016.
9. Y. Liu, X. Sun, and J. H. L. Pang, "A yolov3-based deep learning application research for condition monitoring of rail thermite welded joints," *Proceedings of the 2020 2nd 2020*, pp. 33–38, 2020.
10. K. Bhanot, "Color identification in images," *Towards Data Science*, 2018. [Online]. Available: https://towardsdatascience.com/color-identification-in-images-machine-learning-application-b26e770c4c71
11. R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.
12. T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
13. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
14. C. Rothwell, "Hyper-parameter optimisation using genetic algorithms: Classification task," 2018.
15. E. Wirsansky, *Hands-on genetic algorithms with Python: Applying genetic algorithms to solve real-world deep learning and artificial intelligence problems*. Birmingham, UK: Packt Publishing Ltd, 2020.

Camera-based spatter detection in laser welding with a deep learning approach

Julia Hartung[1,2], Andreas Jahn[1], Martin Stambke[1], Oliver Wehner[1], Rainer Thieringer[1], and Michael Heizmann[2]

[1] TRUMPF Laser GmbH,
Aichhalder Straße 39, 78713 Schramberg
[2] Karlsruher Institut für Technologie,
Institut für Industrielle Informationstechnik,
Hertzstraße 16, 76187 Karlsruhe

Abstract Continuous quality monitoring is essential for automated production systems and efficient manufacturing. Laser welding processes are a key technology for many industrial applications and must fulfill high-quality requirements. Various influencing factors can lead to defects in the weld seam, which impair the functionality and quality of the end product. Therefore, a reliable quality assurance is a prerequisite for high product quality in welding processes. An indicator for an unstable situation in welding processes is the occurrence of spatter on the component. Thus, the detection of spatter can serve as a significant signal for defective weld seams. This article proposes the detection of spatter based on a camera image taken with an industrial camera, which is usually already integrated in the laser system. Due to the large variance of weld seams in image-based analysis, algorithms with a high degree of generalization are required. Using convolutional neural networks (CNN) and semantic segmentation the camera image is analyzed and classified pixel by pixel. The CNN is trained in a multi-class approach in order to recognize the weld seam as well as the spatter as result classes. The segmentation map constitutes the classification result. The results of the deep learning algorithms are evaluated by different methods and conclusions about their prediction quality are made.

Keywords Laser welding, semantic segmentation, u-net, quality assurance, spatter detection

1 Introduction

Laser welding is a key technology in many industrial applications. Due to various advantages, like the possibility to create narrow but deep welding seams and the contactless assembly at highest processing speed, the procedure is more and more used in industry [1, 2]. Remote-controlled laser welding with scanner optics can be integrated as process step in an automated production system and is thus becoming increasingly relevant [3]. To ensure a high welding quality, continuous process monitoring is essential [3,4].

Various influencing factors can lead to defects in the weld seam, which impair the quality and functionality of the end product and often result in safety-relevant risks. In the context of quality assurance, the presence of spatter on the component can be used as an indicator of an unstable situation in the welding process, as its occurrence is closely related to the quality of the weld seam [5, 6]. Spattering is the ejection of melt droplets from a molten bath [4]. There are different types of spatter phenomena that can occur during laser welding. In [7] the formation of spatter and different types of spatter was investigated and a system for categorizing spatter formation was proposed. The effects of droplet ejection from the weld metal can result in a weld seam with underfill, undercuts, craters, blowholes or eruptions that can negatively affect weld properties [7]. Spatter detection therefore serves as a significant signal for defective welds.

As spatters represent height deposits, they can be easily and clearly detected by means of optical coherence tomography (OCT) (figure 3.1a and 3.2a). Just simple image processing algorithms applied on the depth maps such as threshold analysis are necessary. Even if the evaluation of the sensor data is simple and unambiguous, the use of the sensor in this application case has disadvantages. In order to use the OCT sensor, it must first be installed and set up explicitly for quality monitoring on the system. The sensor, which is already expensive to purchase, generates additional effort through calibration procedures and increases the complexity and cost of the overall construction of the system and optics.

By observing the welding process in real time, spatter can be detected as it occurs. In [8], for example, the welding process is monitored by an external high-speed video camera which is sensitive in

the ultraviolet and visible wave length range and captures dynamic images of laser welding plume and spatter directly during the welding. The number and size of the spatters were calculated by using image processing technology and defined as characteristic features. Furthermore, the use of an external high-speed camera for near infrared (IR) measurements was tested. A direct comparison of the images showed, however, that the measurement in UV light and visible light was more suitable for spatter detection [4].

In [9] a setup with a CMOS camera directly at the laser optics is proposed for monitoring the welding process. To get significant images of the weld pool an additional laser for confocal illumination is used and a bandpass filter is placed in front of the camera. Based on the generated images, approaches for scanning the contour of the melt lake and an approach for spatter detection using outlier classification were presented [9].

In comparison to the system setup of the approaches introduced above, an industrial camera is usually already integrated in the system. The camera image is used for example to detect the position of components before welding. However, it is difficult to analyze the weld seams based on images using conventional image processing methods. Even faultless welding seams show a high variance, so that the image processing algorithm for spatter detection must be adapted by experts for each welding process.

Compared to conventional image processing algorithms, deep learning methods tolerate natural deviations in complex patterns. Convolutional neural networks (CNN) offer the advantage that they can be adapted to new procedures without expert knowledge by training procedures, which has already led to very good results. For example in [10] an auto-encoder is used to learn relevant features from the input data. They use a deep neural network to extract salient and low-dimensional features from the high-dimensional laser welding data.

This article proposes the detection of spatter directly after the welding process using the camera image, which does not contain any information about the height profile. Due to the large variance of weld seams in image-based analysis, algorithms with a high degree of generalization are required. The experimental setup is described in section 2 , which is split into the generation and explanation of the

data basis, as well as the analysis and classification of the camera image. In section 2.1, the structure of the neural network is described in more detail and in 2.2 the evaluation methods are further specified. The results are discussed in section 3. A conclusion with a summary of the described algorithms is given in section 4.

2 Experimental setup

The data analysis is performed on 18 mm long weld seams, which connect two sheets with each other. For this study we carried out welding experiments on different materials and with different configurations. The occurrence of spatter as well as the quality of the weld seam depends strongly on the welding parameters. During the experiment we varied the laser power between 4 kW and 6 kW, created a gap between the sheets and induced a defocusing of the scanner optics. This influenced the process in such a way that spatter and also unstable weld seams were produced.

Immediately after welding we took a grayscale camera image with an industrial camera and scanned the height profile of the seam area with an OCT sensor. Both sensors are mounted directly on the welding head and run coaxially with the laser beam path through the beam focusing optics. To get a better camera view, a lighting ring is attached to the scanner optic. By recording both camera and OCT data on a weld seam, the reliable information about the occurrence of spatters can be derived from the height information and used as ground truth. This enables an evaluation of the accuracy of the camera-based prediction, even in cases where the spatters may not be intuitively visible in the camera image.

2.1 Network architecture

A semantic segmentation approach was chosen to evaluate the seam and to recognize spatters in the camera image. The architecture of the convolutional network is based on the u-net architecture [11]. The network learns the structure of the weld seam and the properties of the spatter class in the convolution layers and can thus perform a correct assignment of the image areas. The u-net architecture

relies on the strong use of data augmentation. Data augmentation is essential to teach the network the desired invariance and robustness properties for training with only a few training data sets [11]. Since labeling in semantic segmentation is time-consuming and error prone, it is useful to work with a small amount of training datasets especially in industrial applications. The previously generated data set is enlarged by rotation, vertical and horizontal shift, vertical and horizontal flip, adjustment of the brightness range, zoom and shear, which also improves the robustness of the training. In general, the images were only cut to the seam area during pre-processing and left in their original condition for better performance. Four different classes were defined as output. One class covers the background, another the weld seams welded with optimal parameters, the third class unstable weld seams and the fourth class the spatters.

The network architecture has been reduced in size compared to the original u-net. It is recommended to keep the number of trainable parameters in the architecture low, especially since industrial process images have less variability and less complex properties. In the downsampling the network architecture contains six convolutional layers and three max-pooling layers that each reduce the resolution by a factor of two. Each convolutional layer is followed by an exponential linear unit (ELU), which increases the convergence rate during learning. The ELU was proposed by Clevert et al. [12] as a self-normalizing layer that extends and improves the commonly used ReLU activation. It helps to prevent the Dying-ReLU problem, since it's derivative is different from zero for negative values. Several other studies have shown improvements in training and results as well. Our tests confirm these results, which is why we use the ELU function in the network architecture. The number of feature channels is doubled per downsampling step similar to the original u-net architecture. After the corresponding upsampling a final layer with a 1x1 convolution followed by a softmax activation is used to map each feature vector to the desired number of classes.

The model is not pretrained, but the Xavier Glorot uniform initialization method is used to initialize the weights [13].

2.2 Evaluation

Training approaches with two different loss functions were evaluated. The first approach uses the weighted categorical cross entropy loss (WCCE) and the other the weighted dice coefficient loss function.

Since the number of pixels per class is very different and especially the less important background class contains most pixels, the weighting in the loss function generates better results. The pixel ratio values of the different classes are as follows: background: 82%, seam welded with optimal parameter: 6.3%, unstable weld seam: 11.6% and spatter 0.1%. In comparison, the frequency of occurrence of the classes in all images is follows: background: 100%, stable weld seam: 33.3%, unstable weld seam: 66.6% and spatters 86%.

The class weight has been defined to give priority to the evaluation of the weld seam and also to force the detection of spatter. We choose a weighting of 0.1 for the background, 0.25 each for stable weld seam and unstable weld seam and 0.4 for the spatter class. Attention must be paid to ensure that the weighting does not penalize the most common class (background) too much, otherwise some pixels will no longer be classified. Therefore a good ratio for the weightings must be found.

The neural network was trained with a training data set of 251 images. 74 images are of weld seams welded with optimal parameters, while the other 177 images show weld seams that establish the different defect classes. For labeling the camera images depth data on basis of the OCT data are used as ground truth. With the knowledge of the height information all spatters can be recognized and labeled. The weld seams and spatters were marked (optimally welded seams in green, defective weld seams in blue and spatters in red, see figure 3.1c and figure 3.2c). With a good setting of the laser parameters, far fewer spatters are produced than with poorly selected parameters. Therefore, spatter occurs more often with defective welds than with good ones.

A quarter of the training data was used as validation data set.

3 Results

After training of 184 epochs with 150 steps per epoch, a batch size of 20 and the use of the weighted dice coefficient loss function a training error of 0.13 and validation error of 0.23 was achieved. The weighted dice coefficient loss function provides better results than the WCCE approach.

To evaluate the result the weighted dice coefficient loss, also known as the Sørensen-dice coefficient or F1 score, is used too. Therefore, we use the function

$$\text{Loss function} = 1 - \text{dice} \tag{3.1}$$

with

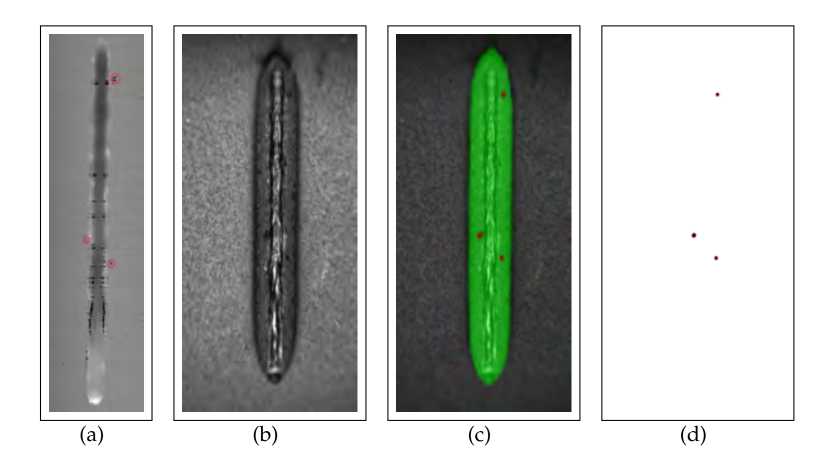
$$\text{dice} = \frac{2 \mid X \cap Y \mid}{\mid X \mid + \mid Y \mid} \tag{3.2}$$

where $\mid X \mid$ and $\mid Y \mid$ are the cardinalities of the two sets.

The dice value is calculated for each individual class, weighted with the respective class weighting and then added up.

The loss value on the test data set is 0.27. If only the spatter class is taken into account, a loss value of 0.32 is achieved. It must be considered that the spatters contain only very few pixels compared to the total image and that these cannot always be labeled exactly on basis of the ground truth.

Figure 3.1 and figure 3.2 show examples of segmentation maps predicted by the neural network trained with the weighted dice coefficient loss function. In both examples the spatters were detected, and the welding seam was correctly classified as being welded with optimal parameters or as a weld of poorer quality. In figure 3.1(a) the weld seam and three spatters are shown in an image generated from the height profile of the OCT sensor. In figure 3.1(b) the corresponding camera image of the same weld with spatter is shown. The grayscale image is analyzed using a deep learning approach and classified with pixel-level semantic segmentation according to weld seam and spatter. The result is shown in figure 3.1(c). The detected

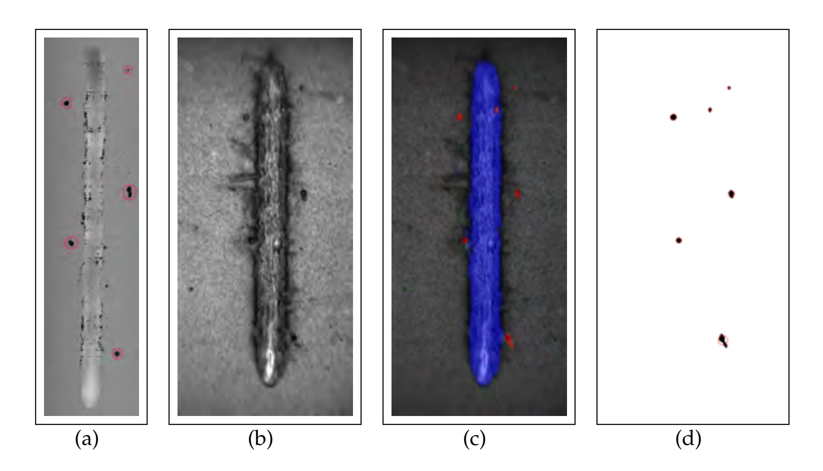


(a) (b) (c) (d)

Figure 3.1: (a) image generated on the depth data on basis of the OCT sensor, (b) camera image, (c) overlay image with the predicted segmentation map, (d) detected spatters

seam is labeled in green, while the spatters are labeled in red. Figure 3.1(d) shows the detected spatters counted for the comparison with the ground truth. Corresponding to figure 3.1 the different pictures of an error seam are shown in 3.2. In this case the detected seam is labeled in blue, because it is a weld seam of poor quality.

However a better comparison is provided by the number of detected spatters in the image compared with the number of spatters in the ground truth. In this evaluation approach the precisely labeled pixels are not important, only the amount of detected spatters is taken into account. With a test data set of 102 images, an average deviation of 0.41 spatters per image was observed. The number of spatters was correctly detected in 77 of the images. In the other cases either not all spatters were detected or discoloration in the sheet or on the welding seam was also classified as spatter. The ratio between the two error cases is quite balanced. The highest deviations were caused by the test sets containing many spatters. With 5 misclassifications, these are very significant in the average result value. In figure 3.3 the classification result of the test data set is shown in a more detailed way. The number of spatter in the ground truth is



(a) (b) (c) (d)

Figure 3.2: (a) image generated on the depth data on basis of the OCT sensor, (b) camera image, (c) overlay image with the predicted segmentation map, (d) detected spatters

compared to the number of spatter in the prediction. The two cases in which 5 spatter were not detected are shown in the bottom two lines at ground truth 10 and 11. But more decisive for the weld seam evaluation are the cases in which a picture is classified as spatter-free despite spatter in the ground truth, or the other way round in which spatters are detected in a picture that has no spatter in the ground truth. These cases would lead to false conclusions about the seam quality and should therefore be avoided. In our test data set spatters were classified on two images although there were none in the ground truth and once no spatter was found on the test image although the ground truth indicated two small spatters. These values are shown in figure 3.3 at ground truth 0 and prediction 1 and the other case at ground truth 2 and prediction 0.

In our Test dataset of the 102 test images, too few spatters were detected on 13 images and too many spatters on 12 images.

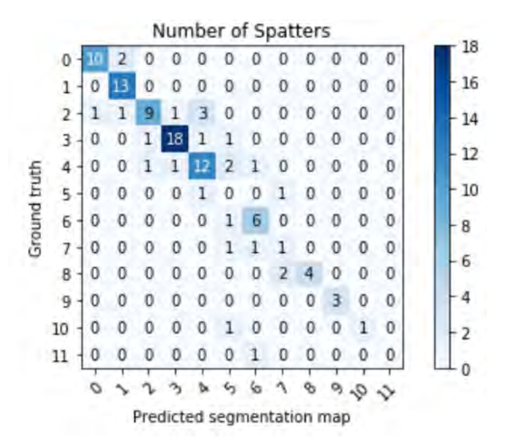


Figure 3.3: Comparison of the number of spatter in the ground truth and the prediction in the test data set

4 Conclusion

In this paper an approach for spatter detection in a laser welding process with an industrial camera was presented. This was achieved by using a semantic segmentation approach to slim down the image features and classify the image pixel by pixel. Even with a small training data set all spatters could be correctly classified on 75% of the images from the test data set. Only on 3 out of 102 test images no spatter was detected in spite of existing spatters in the ground truth, or spatter was detected on images that actually contained no spatter. This results in an effective error rate with wrong conclusion of 2.9%. This result proves that quality monitoring is possible with a simple system setup. The setup of a fixed industrial camera is mostly standard in laser welding due to seam position control or other functions required for welding. This means that process monitoring can be done without additional hardware and the resulting costs or installation work. This aspect should not be ignored when implementing a system in industry. In addition, neither high-resolution images nor complex pre-processing algorithms were used, which would require longer processing time and higher computing power. Promising results were achieved on the industrial data set,

which justifies an image-based quality assessment using deep learning in the industrial environment.

References

1. M. Jäger, S. Humbert, and F. Hamprecht, "Sputter Tracking for the Automatic Monitoring of Industrial Laser-Welding Processes," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 2177–2184, 2008.
2. A. Bollig, S. Mann, R. Beck, and S. Kaierle, "Einsatz optischer Technologien zur Regelung des Laserstrahlschweißprozesses," in *Autom.*, 2005.
3. M. Zaeh, J. Moesl, J. Musiol, and F. Oefele, "Material processing with remote technology revolution or evolution?" *Physics Procedia*, vol. 5, pp. 19 – 33, 2010, laser Assisted Net Shape Engineering 6, Proceedings of the LANE 2010, Part 1.
4. D. You, X. Gao, and S. Katayama, "Visual-based spatter detection during high-power disk laser welding," *Optics and Lasers in Engineering*, vol. 54, pp. 1 – 7, 2014.
5. A. Kaplan and J. H. Powell, "Laser welding: The spatter map," *International Congress on Apllications of Laser & Electro-Optics*, pp. 683–690, 2010.
6. M. Zhang, G. Chen, Y. Zhou, S. Li, and H. Deng, "Observation of spatter formation mechanisms in high-power fiber laser welding of thick plate," *Applied Surface Science*, vol. 280, pp. 868 – 875, 2013.
7. A. Kaplan and J. H. Powell, "Spatter in laser welding," *Journal of Laser Applications*, vol. 23, pp. 1–7, 2011.
8. X. Gao, Y. Sun, and S. Katayama, "Neural network of plume and spatter for monitoring high-power disk laser welding," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 1, pp. 293 – 298, 2014.
9. N. C. Stache, H. Zimmer, J. Gedicke, A. Olowinsky, and T. Aach, "Robust High-Speed Melt Pool Measurements for Laser Welding with Sputter Detection Capability," in *DAGM07: 29th Annual Symposium of the German Association for Pattern Recognition*, ser. LNCS, F. A. Hamprecht, C. Schnörr, and B. Jähne, Eds., vol. 4713. Heidelberg: Springer, Sept. 12–14 2007, pp. 476–485.
10. J. Günther, P. M. Pilarski, G. Helfrich, H. Shen, and K. Diepold, "Intelligent laser welding through representation, prediction, and control learning: An architecture with deep neural networks and reinforcement

learning," *Mechatronics*, vol. 34, pp. 1 – 11, 2016, system-Integrated Intelligence: New Challenges for Product and Production Engineering.

11. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *CoRR*, vol. abs/1505.04597, 2015.
12. D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," *CoRR*, vol. abs/1511.07289, 2016.
13. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010.

Semantische Segmentierung von Ankerkomponenten von Elektromotoren

Norbert Mitschke und Michael Heizmann

Karlsruher Institut für Technologie,
Institut für Industrielle Informationstechnik,
Hertzstraße 16, 76187 Karlsruhe

Zusammenfassung In diesem Beitrag wird die semantische Segmentierung von Ankern aus Elektromotoren und seinen Komponenten untersucht. Hierfür wird ein U-Net mit einem eigenständig angefertigten Datensatz trainiert, welcher aus Bildern von Ankern unterschiedlichster Bauformen besteht und im Rahmen dieses Beitrags angefertigt wurde. Aufgrund der geringen Anzahl von 75 Trainingsbildern werden neben einer geeigneten Standardaugmentierung auch eine neuartige Hintergrundaugmentierung und das Einbinden von Kanteninformationen untersucht. Mithilfe dieser Methoden kann der Testfehler bei der Segmentierung um insgesamt 70% reduziert werden.

Keywords Neuronale Netze, maschinelles Lernen, semantische Segmentierung, automatische Sichtprüfung

1 Einleitung

In diesem Beitrag wird ein Ansatz für die semantische Segmentierung der Komponenten von Altprodukten am Beispiel des Ankers von Elektromotoren vorgestellt. Die Segmentierung der Komponenten stellt den ersten Schritt für die Rückgewinnung von Altprodukten, dem sog. *Remanufcturing*, dar. Hierfür ist es erforderlich, die funktionsrelevanten Komponenten des Altproduktes zu erkennen, um diese anschließend inspizieren zu können. An die Erkennung ist eine hohe Anforderung an die Genauigkeit gebunden, da in weiteren Arbeiten auf Basis des Segmentierungsergebnisses nicht nur

die Lageparameter geschätzt werden, sondern das Ergebnis auch als Maske für das Zusammensetzen der Mantelfläche (engl. *stitching*) verwendet wird.

Somit ist ein möglichst robuster Klassifikator auf Pixelebene erforderlich, der einerseits Anker mit ungewissen Produktzuständen, die beispielsweise durch Defekte gegeben sind, und andererseits verschiedenste Ankerbauformen erkennt. In der Vergangenheit haben sich neuronale Netze [1] als vorteilhaft für komplexe Bildverarbeitungsaufgaben wie Klassifikation, Detektion oder Segmentierung herausgestellt. Speziell für die semantische Segmentierung von Bildern ist die Verwendung eines U-Net [2] der Stand der Technik.

Zunächst wird in Abschnitt 2 der Datensatz präsentiert, der für diesen Beitrag erstellt wurde. Anschließend wird in Abschnitt 3 der verwendete Ansatz vorgestellt. Dieser umfasst die Augmentierung in Abschnitt 3.1 und die Erweiterungen des U-Net in Abschnitt 3.3. Anhand des beschriebenen Versuchsaufbaus in Abschnitt 4 werden in Abschnitt 5 die Ergebnisse beschrieben. Der Beitrag schließt mit einer Zusammenfassung in Abschnitt 6.

2 Datensatz

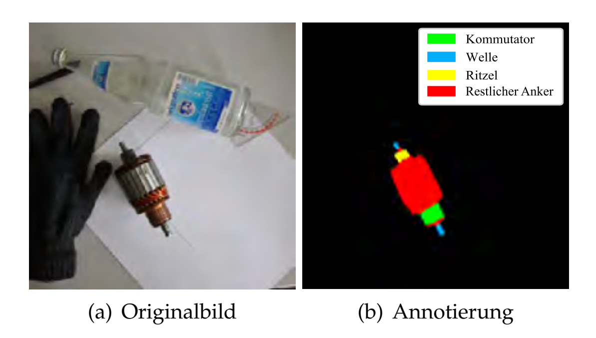
Das Lernen eines neuronalen Netzes erfordert eine Vielzahl an annotierten Bildern mit geeignetem Kontext. Für die Zwecke der Segmentierung von Ankern in Elektromotoren ist bisher kein öffentlich zugänglicher Datensatz verfügbar, weswegen ein relativ kleiner Datensatz mit insg. 96 Bildern erstellt wurde, da das Annotieren mit einem hohen Zeit- und Kostenaufwand verbunden ist. Der Datensatz wird im Folgenden beschrieben.

Der Datensatz besteht einerseits aus selbst aufgenommenen Bildern der am Institut vorliegenden Anker und andererseits aus frei zugänglichen Bildern aus dem Internet. Die eigenen Aufnahmen haben verschiedene irrelevante Objekte im Hintergrund, während die Bilder aus dem Internet oft von Online-Händlern stammen und einen einfarbigen Hintergrund aufweisen. Um die Bilder als Eingang für das neuronale Netz verwenden zu können, werden diese zu einem Quadrat beschnitten und anschließend mit einem geeigneten Aliasing-Filter auf 224×224 Pixel herunter- bzw. heraufgetastet. Von

den 96 Bildern werden 75 als Lernbilder und 21 als Testbilder verwendet.

Für das *Remanufcturing* sind drei Ankerkomponenten relevant. Diese sind der Kommutator (K), die Welle (W) und das Ritzel (R), deren Leitfähigkeit bzw. mechanische Eigenschaften starken Einfluss auf die Funktionsfähigkeit des Motors haben. Bei der Annotierung erhält jedes Pixel die Information, ob es zum Anker gehört und ggf. zu welcher Klasse es gehört (s. Abb. 2.1). Es ergibt sich somit ein Vier-Klassen-Problem innerhalb der Ankermaske, das die drei relevanten Klassen und eine Dummy-Klasse (X) enthält. Letztere beschriebt den restlichen Anker, d. T. Teile des Ankers, die keiner oben genannten Klasse zuzuordnen sind. Für die Menge aller Pixel des Ankers A und der Klassen K, W, R und X gilt

$$\begin{aligned}
&\mathrm{K}, \mathrm{W}, \mathrm{R}, \mathrm{X} \in \mathrm{A}, \\
&\mathrm{P} \cap \mathrm{Q} = \begin{cases} \emptyset, & \mathrm{P} \neq \mathrm{Q}, \\ \mathrm{P}, & \mathrm{P} = \mathrm{Q} \end{cases} \\
&\qquad \text{für } (\mathrm{P}, \mathrm{Q}) \in (\{\mathrm{K}, \mathrm{W}, \mathrm{R}, \mathrm{X}\} \times \{\mathrm{K}, \mathrm{W}, \mathrm{R}, \mathrm{X}\}), \\
&\mathrm{K} \cup \mathrm{W} \cup \mathrm{R} \cup \mathrm{X} = \mathrm{A}.
\end{aligned} \tag{2.1}$$



(a) Originalbild (b) Annotierung

Abbildung 2.1: Bei der Annotierung wird im Originalbild nach dem Anker und seinen Komponenten (Kommutator, Welle und Ritzel) gesucht.

3 Ansatz

In diesem Beitrag werden zwei Ansatzpunkte für die Verbesserung der Robustheit eines Detektors untersucht. Zum einen wird die Variationen der Anker durch Augmentierung beim Training erhöht. Dies kann als integrative Methode zur Erzeugung invarianter Merkmale verstanden werden [3]. Zum anderen wird Vorwissen verwendet, um triviale Fehler beim Lernen zu vermeiden und so das Training zu beschleunigen. Beide Ansätze werden im Folgenden für den in Abschnitt 2 beschriebenen Datensatz untersucht.

3.1 Augmentierung

Da die Annotierung von Bilddaten oft sehr zeit- oder kostenintensiv ist, sind die Lerndatensätze oft sehr klein, was in der Lernphase zu Überanpassung führt. Neben Regularisierung, Dropout und Batch-Normalisierung wird auch Bildaugmentierung verwendet. Hierbei wird eine Transformation auf ein Bild ausgeführt, die die Bildelemente manipuliert, während die Annotierung nur kohärent beeinflusst wird.

Für eine Segmentierung kommen fünf Arten von Augmentierung infrage: Spiegelung, affine Transformationen, Farbmanipulationen, Rauschen und Cutout. Hierbei müssen die Operationen an die Ankerbilder angepasst werden und können teilweise erweitert werden.

Eine zentrale Rolle spielt die Skalierung bei der affinen Transformation, da sie die Größe des Objektes bestimmt. Da die Bilder des Datensatzes Anker unterschiedlichster Größe enthalten, muss die Skalierung abhängig vom Bild so gewählt werden, dass die resultierende Größe des Ankers im Bild innerhalb einer gewissen Schwankungsbreite liegt. In diesem Betrag wird als Schwankungsbreite 10% bis 40% der Bildgröße gewählt, was ca. 5.000 bis 20.000 Pixeln entspricht.

Für alle Augmentierungsoperationen werden die Parameter stochastisch in geeigneten Grenzen gewählt. In den Experimenten wird eine sechsstufige Augmentierungspipeline verwendet, die aus den folgenden Stufen besteht:

- Spiegelung (keine, x-Achse, y-Achse, x- und y-Achse)

- Rotation ($-\frac{\pi}{2}$ bis $+\frac{\pi}{2}$)
- Skalierung durch Ausschneiden oder Padding (s. oben)
- Translation und Scherung entlang x- und y-Achse
- Cutout nach [4]
- Gaußfilter, Schärfung, Rauschen, Änderungen von Helligkeit bzw. Sättigung, Farbwertquantisierung oder Farbverschiebung

Da in den Bilder des Lerndatensatzes meist ein Anker als einziges Objekt vorhanden ist, besteht die Gefahr, dass das U-Net nur das Vorhandensein eines bloßen Objektes erlernt und es somit nicht vom spezifische Objekt, dem Anker, unterscheiden kann. Um dies zu vermeiden, wird Hintergrundaugmentierung verwendet. Hierzu wird die Grundwahrheit als binäre Maske $\mathbf{m}_\mathbf{I}$ zur Extraktion des Ankers aus dem Bild $\mathbf{I}$ benutzt. Anschließend wird der extrahierte Anker vor einen zufälligen Hintergrund $\mathbf{H}_i$ aus dem dtd-Datensatz [5] gelegt. Das Ergebnis wird anschließend mit dem Tiefpassfilter $\mathbf{g}_\mathbf{Gauß}$ gefiltert. Es ergibt sich

$$\mathbf{I}_\mathbf{aug} = \mathbf{g}_\mathbf{Gauß} ** (\mathbf{m}_\mathbf{I} \odot \mathbf{I} + (1 - \mathbf{m}_\mathbf{I}) \odot \mathbf{H}_i). \tag{3.1}$$

3.2 U-Net

Das U-Net nach [2] ist in Abb. 3.1 illustriert. Die Eingabe ist ein RGB-Bild und die Ausgabe gibt die geschätzte Klassenzugehörigkeit für jedes Pixel an. Die namensgebende Form des neuronalen Netzes entsteht durch die kleiner, aber dafür tiefer werdenden Merkmalskarten zur Mitte hin und die Querverbindungen, bei denen Merkmalskarten gleicher Größe konkateniert werden (gelbe Pfeile in Abb. 3.1).

Das verwendete U-Net hat eine Eingabegröße von 224×224 Pixeln, fünf Tiefenstufen und ca. 31 Mio. trainierbare Parameter. Auf jede 3×3-Faltungsschicht folgt Batch-Normalisierung nach [6] und eine ReLU-Aktivierung. Beim Hochtasten wird das 2×2-Interpolationsfilter auch im Training gelernt.

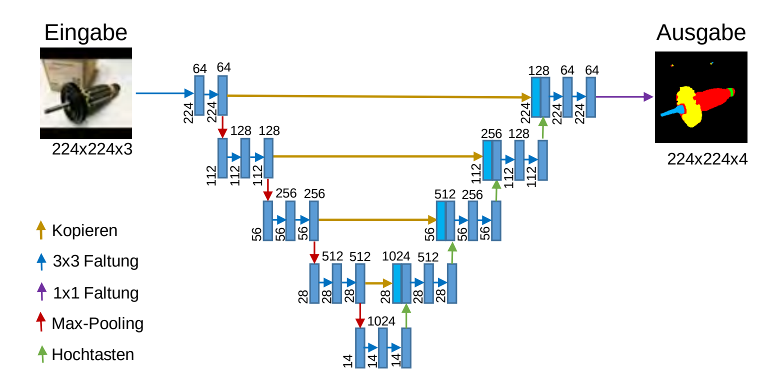


Abbildung 3.1: Die Abbildung zeigt den hier verwendeten Aufbau des U-Nets. Die oberen Zahlen an den blauen Rechtecken geben Anzahl der Merkmale bzw. die Tiefe der Aktivierungskarten an, während die seitlichen Zahlen die Höhe bzw. die Breite der Aktivierungskarte wiedergeben.

Als Zielfunktion wird der generalisierte Sörensen-Dice-Koeffizient c_{gSDK} nach [7] verwendet. Dieser bildet die gewichtete Summe der Sörensen-Dice-Koeffizienten oder einzelnen Klassen. Es gilt:

$$c_{\text{gSDK}} = \sum_i w_i \cdot c_i = \sum_i w_i \cdot \left(1 - \frac{|\mathbf{y} \cap \hat{\mathbf{y}}|}{|\mathbf{y} \cup \hat{\mathbf{y}}|}\right). \tag{3.2}$$

Mit dem Sörensen-Dice-Koeffizienten wird das negative Verhältnis von Schnitt zur Vereinigung zweier Flächen abgebildet. Nähert sich der Sörensen-Dice-Koeffizient dem Wert 0, so sind der Schnitt und die Vereinigung identisch.

3.3 Erweiterung des U-Net

Neben Augmentierung eignen sich zusätzliche Informationen, um die Genauigkeit des neuronalen Netzes zu erhöhen. In diesem Abschnitt werden Methoden aufgeführt, die das U-Net um Zusatzinformation erweitern.

Die Hinzunahme der Kanteninformation in einer der hinteren Schichten kann zu einer Verbesserung führen, da die Objektgrenzen des Ankers im Bild mit den Kanten im Bild zusammenfallen.

Zur Kantenextraktion wird der Marr-Hildreth-Operator verwendet. Das Ergebnis wird anschließend normiert. Die Kante wird nach der obersten Konkatenierungsschicht entweder hinzuaddiert oder angehängt. Die schematische Veränderung des U-Net ist in Abb. 3.2 dargestellt.

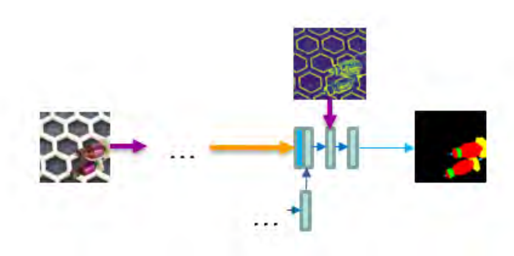


Abbildung 3.2: Die Abbildung zeigt, wie die letzten Schichten des U-Net abgeändert werden, um die Kanteninformationen einzubringen.

Da es sich bei den Ankern um zusammenhängende Objekte handelt, ist eine Regularisierung sinnvoll, die lange Konturen bestraft. Somit können Löcher, kleine Fehldetektionen oder andere Artefakte reduziert werden. Hierfür eignet sich die *Total-Variation*-Regularisierung (TV-Regularisierung) nach [8]. Der Strafterm L_{TV} für das Segmentierungsergebnis $\mathbf{A} = [\mathbf{a}^{(\text{A})}, \mathbf{a}^{(\text{K})}, \mathbf{a}^{(\text{W})}, \mathbf{a}^{(\text{R})}] \in \mathbb{R}^{(224,224,4)}$, wobei das x in $a^{(\text{x})}$ für die Aktivierungskarte des Ankers (A), des Kommutators (K), der Welle (W) oder des Ritzels (R) steht, wird gewählt zu

$$L_{\text{TV}} = \lambda_{\text{TV}} \sum_{n=\{\text{A,K,W,R}\}} w_n \cdot$$
$$\sum_i \sum_j \mathbf{a}^{(n)}[i,j] - \frac{1}{2}\mathbf{a}^{(n)}[i+1,j] - \frac{1}{2}\mathbf{a}^{(n)}[i,j+1]. \tag{3.3}$$

Eine Erweiterung hiervon ist, den Strafterm an Stellen, an denen eine Kante vorliegt, zu verkleinern.

Im originalen U-Net wird das Ergebnis nach der letzten Faltungsschicht mit einer Sigmoid-Funktion $\sigma(\,\cdot\,)$ aktiviert. Dies kann potentiell dazu führen, dass sich die Klassen nicht gegenseitig ausschließen oder einzelne Teile einer Komponente wie bspw. der Welle zwar als

Welle erkannt werden, aber nicht als Teil des Ankers erkannt werden. Dies kann durch zusätzliche Restriktionen vermieden werden, die allerdings die Konvergenzeigenschaften des Netzes beeinflussen. Im Folgenden werden zwei Alternativen vorgestellt, das Ergebnis der letzten Faltungsschicht $\mathbf{A}$ zu aktivieren.

Der erste Ansatz ist ein multiplikativer Ansatz mit Sigmoid-Aktivierung (MSig), der auschließt, dass Komponenten außerhalb der Ankerklasse liegen. Die Aktivierungsvorschrift für die Klasse x lautet

$$\begin{aligned}&\text{für } x = A\text{: } \mathbf{b}^{(\mathrm{A})} = \sigma(\mathbf{a}^{(\mathrm{A})})\\&\text{für } x \neq A\text{: } \mathbf{b}^{(x)} = \mathbf{b}^{(\mathrm{A})} \cdot \sigma(\mathbf{a}^{(x)}).\end{aligned} \tag{3.4}$$

Beim zweiten Ansatz (MSmax) wird das gegenseitige Ausschließen der Klassen durch eine Softmax-Funktion S sichergestellt. Es ergibt sich

$$\begin{aligned}&\text{für } x = A\text{: } \mathbf{b}^{(\mathrm{A})} = \sigma(\mathbf{a}^{(\mathrm{A})})\\&\text{für } x \neq A\text{: } \mathbf{b}^{(x)} = \mathbf{b}^{(\mathrm{A})} \cdot S([\mathbf{a}^{(x)}, \mathbf{0}]).\end{aligned} \tag{3.5}$$

Für den Fall eines Pixels im Anker ohne Zugehörigkeit zur Klasse K, W oder R wird für MSmax eine Aktivierungskarte mit dem konstanten Wert 0 hinzugefügt. Dies entspricht der Klasse X in Abschnitt 2.

4 Versuchsaufbau

Jedes Einzelexperiment wird dreimal wiederholt. Das Ergebnis wird gemittelt. Bei jedem Durchlauf wird das U-Net für 100 Durchläufe zu je 16×1024 augmentierten Bildern mit dem *Nadam*-Optimierer trainiert. Es wird ein kosinusartiger Rückgang der Lernrate mit einer Anfangslernrate von 10^{-3} verwendet. Als Vergleichsmetrik wird der generalisierte Sörensen-Dice-Koeffizient und der Jaccard-Koeffizient der einzelnen Komponenten verwendet.

Zuerst wird der Einfluss von Augmentierung untersucht. Es werden vier Stufen der Augmentierung mit und ohne Hintergrundaugmentierung untersucht. Bei der Hintergrundaugmentierung wird

mit der Wahrscheinlichkeit 0,6 ein zufälliger Hintergrund verwendet, ansonsten bleibt der Originalhintergrund bestehen. In der untersten Stufe (Stufe 0) wird keine Augmentierung durchgeführt. In Stufe I wird die Basisaugmentierungskaskade verwendet, die aus Spiegelung, Rotation, Skalierung, Translation, Scherung und Cutout besteht. Für Stufe II wird diese Kaskade gemäß Abschnitt 3.1 um eine Stufe mit den Operationen des letzten Stichpunktes von Abschnitt 3.1 erweitert. In der letzten Stufe (Stufe III) wird die Cutout-Stufe um zwei eigene Verfahren erweitert. Zum einen werden zufällig einzelne Komponenten verdunkelt und zum anderen wird Cutout mehrfach mit kleineren Rechtecken angewendet.

Danach werden mit der besten Augmentierungsstrategie die Verfahren aus Abschnitt 3.3 verglichen. Zunächst wird der Einfluss der Aktivierung der letzten Schicht untersucht. Anschließend werden verschiedene Kombinationen aus TV-Regularisierung und Hinzufügen von Kanteninformationen betrachtet.

5 Ergebnisse

Im Folgenden werden die Ergebnisse für die Augmentierung und für Erweiterungen des U-Net vorgestellt.

5.1 Augmentierung

Die Ergebnisse sind in Tabelle 1 dargestellt und zeigen, dass sich der Sörensen-Dice-Koeffizient bei der Verwendung eines zufälligen Hintergrunds bei allen Augmentierungsstufen um ca. 40% verbessert. Dies stützt die These, dass durch einen zufälligen Hintergrund der Fokus des Trainings auf das relevante Objekte verlagert wird, woraus eine bessere Generalisierung des Netzes folgt. Ohne Augmentierung des Objekts führt Hintergrundaugmentierung zu einer Verschlechterung von 36%, da die Position des Objektes vom Netz auswendig gelernt werden kann.

Bei der hier getroffenen Auswahl der Augmentierungskaskade führt eine stärkere Augmentierung zu leicht besseren Sörensen-Dice-Koeffizienten. Daher wird das beste Segmentierungsergebnis bei Stufe III mit Hintergrundaugmentierung erzielt.

Tabelle 1: Ergebnisse der Augmentierung. Es ist der generalisierte Sörensen-Dice-Koeffizient des Gesamtergebnisses sowie der Jaccard-Koeffizient der Komponenten angeben.

	III	II	I	0	III	II	I	0
	Originaler Hintergrund				Zufälliger Hintergrund			
gSDK	0,179	0,178	0,188	0,328	**0,100**	0,102	0,103	0,447
Anker	0,885	0,887	0,886	0,762	**0,952**	0,951	**0,952**	0,639
Kommutator	0,619	0,616	0,595	0,389	0,650	**0,656**	0,657	0,368
Welle	0,457	0,452	0,451	0,271	**0,560**	0,552	0,548	0,243
Ritzel	0,359	0,357	0,316	0,247	0,461	**0,462**	0,455	0,179

5.2 Erweiterung des U-Nets

Für die Standardaktivierung ergibt sich ein gSDK von 0,100. Die beiden anderen Aktivierungen liefern ein gSDK von ebenfalls 0,100 (MSig) bzw. von 0,105 (MSmax). Trotz der Beseitigung aller logischen Widersprüche verschlechtert sich das Ergebnis. Für die weitere Analyse werden daher nur die Sigmoid-Aktivierung und MSig miteinander verglichen, auch weil für bestimmte Kombinationen von MSmax der Jaccard-Koeffizient der Welle nicht konvergiert. Insgesamt zeigt sich, dass die logischen Zusammenhänge beim Training eigenständig erlernt werden.

Tabelle 2: Ergebnisse bei Verwendung der Zusatzinformationen. Mit *K* ist die Konkatenierung und mit *A* die Addition der Kanten gemeint. *R* bedeutet reguläre TV-Regularisierung und *G* die mit Kanten gewichtete. Bei – wird keine Kanteninformation bzw. TV-Regularisierung verwendet.

	Sigmoid-Aktivierung								
Kante	-	-	-	K	K	K	A	A	A
TV-Reg.	-	R	G	-	R	G	-	R	G
gSDK	0,100	0,101	0,104	**0,096**	0,102	0,104	0,100	0,102	0,103
	MSig								
Kante	-	-	-	K	K	K	A	A	A
TV-Reg.	-	R	G	-	R	G	-	R	G
gSDK	0,100	0,101	0,100	0,099	0,102	0,100	0,097	0,102	0,099

Die Ergebnisse sind in Tab. 2 zusammengefasst. Insgesamt sind die erzielten Verbesserungen mit bis zu 5% eher moderat. Eine TV-

Regularisierung hat eher negative Auswirkungen auf das Ergebnis, während Kanteninformationen neutrale bis positive Auswirkungen haben. Am besten schneidet das Verfahren mit Sigmoid-Aktivierung und Kantenkonkatenierung ab.

6 Zusammenfassung

In diesem Beitrag wird ein Segmentierungsnetz für Anker von Elektromotoren vorgestellt, bei dem relevante Komponenten vom Rest des Ankers und dem Hintergrund getrennt werden, um diese anschließend inspizieren zu können. Durch Augmentierung und insbesondere der hier vorgestellten Hintergrundaugmentierung kann das Ergebnis signifikant verbessert werden. Mithilfe von Kanteninformationen kann die Genauigkeit um weitere 4% erhöht werden.

Mit den erzielten Ergebnissen können im Anschluss Lageparameter wie Rotation oder perspektivische Verzerrung des Ankers geschätzt werden. Dies ermöglicht eine bildbasierte Regelung für die optimale Ausrichtung einer positionierbaren Kamera und eine Extraktion der Mantelfläche der relevanten Komponenten.

In weiteren Arbeiten soll das U-Net deutlich länger mit den gefundenen Parametern angelernt und anschließend für die Segmentierung von Videos bzw. Echtzeit-Kamerasystemen verwendet werdet. Mithilfe eines internen Modells soll das Segmentierungsergebnis stabilisiert werden und die Größe des Ankers im jeweiligen Eingabebild durch *Zero-Padding* oder Heranzoomen auf die im Training festgelegte Größe geregelt werden.

Danksagung

Das Projekt AgiProbot wird durch die Carl-Zeiss-Stiftung gefördert.

Literatur

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

2. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.
3. H. Schulz-Mirbach, "Constructing Invariant Features by Averaging Techniques," in *12th IAPR International Conference on Pattern Recognition, Conference B: Patern Recognition and Neural Networks, ICPR 1994, Jerusalem, Israel, 9-13 October, 1994, Volume 2*, 1994, pp. 387–390. [Online]. Available: https://doi.org/10.1109/ICPR.1994.576950
4. T. Devries and G. W. Taylor, "Improved Regularization of Convolutional Neural Networks with Cutout," *CoRR*, vol. abs/1708.04552, 2017. [Online]. Available: http://arxiv.org/abs/1708.04552
5. M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing Textures in the Wild," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.
6. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.
7. R. Crum, O. Camara, and D. Hill, "Generalized Overlap Measures for Evaluation and Validation in Medical Image Analysis," *IEEE Transactions on Medical Imaging*, vol. 25, no. 11, pp. 1451–1461, 2006.
8. D. Strong and T. Chan, "Edge-Preserving and Scale-Dependent Properties of Total Variation Regularization," *Inverse Problems*, vol. 19, no. 6, p. S165, 2003.

Comparing Optimization Methods for Deep Learning at the Example of Artistic Style Transfer

Alexander Geng[1], Ali Moghiseh[2], Katja Schladitz[2], and Claudia Redenbach[1]

[1] University of Kaiserslautern, Gottlieb-Daimler-Straße 47, 67663 Kaiserslautern
[2] Fraunhofer Institute for Industrial Mathematics ITWM, Fraunhofer-Platz 1, 67663 Kaiserslautern

Abstract Artistic style transfer is an application of deep learning using convolutional neural networks (CNN). It combines the content of one image with the style of another one using so-called perceptual loss functions. More precisely, the training of the network consists in choosing the weights such that the perceptual loss is minimized. Here, we study the impact of the choice of the optimization method on the final transformation result. Training an artistic style transfer network with several optimization methods commonly used in deep learning, we obtain significantly differing models. In a default parameter setting, we show that Adam, AdaMax, Adam_AMSGrad, Nadam, and RMSProp yield better results than AdaDelta, AdaGrad or RProp, both measured by the perceptual loss function and by visual perception. The results of the last three methods strongly depend on the chosen parameters. With a suitable selection, AdaGrad and AdaDelta can achieve results similar to the versions of Adam or RMSProp.

Keywords Convolutional neural network, perceptual loss, stochastic gradient descent

1 Introduction

In order to achieve artistic style transfer as first described in [1,2] in real time as desirable for live demonstrations, we train a feed forward network for style images and transfer the styles to content images as specified in [3]. The training of such a network is essentially an optimization problem. The weights of the network are chosen such that the prediction is as close as possible to the training data. We compare eight methods available in PyTorch [4]: AdaGrad [5], AdaDelta [6], RProp [7], RMSProp [8] and the four variants of Adam (Adam, AdaMax, Adam_AMSGrad [9], Nadam [10]).

2 Method

An overview of the system is visualized in Figure 2.1. It consists of two components: an *image transformation network* f_W on the left and a *loss network* ϕ on the right side, which is used to define several *loss functions*. The mapping $\hat{y} = f_W(x)$ transforms the input image x into the output image $\hat{y}$, where W are the weights of the image transformation network. We consider loss functions $\ell_{feat}(\hat{y}, y_1)$ and $\ell_{style}(\hat{y}, y_2)$ which measure the content and style differences between the transformed image $\hat{y}$ and the *target images* y_1 (content target) and y_2 (style target), respectively. In our case, the content target image y_1 is the same as the input image x. Training of the image transformation network consists in minimizing a weighted combination of the loss functions by using a suitable optimization method. We get an optimal value W^* with

$$W^* = \arg\min_W \left[\lambda_c \ell_{feat}(f_W(x), y_1) + \lambda_s \ell_{style}(f_W(x), y_2) \right], \quad (2.1)$$

where λ_c and λ_s are non-negative weight factors.

2.1 Image transformation network

The image transformation network is a deep convolutional neural network consisting of several convolutional layers with varying stride. All convolutional layers are followed by spatial batch normalization and rectified linear units (ReLU) cutting off negative parts.

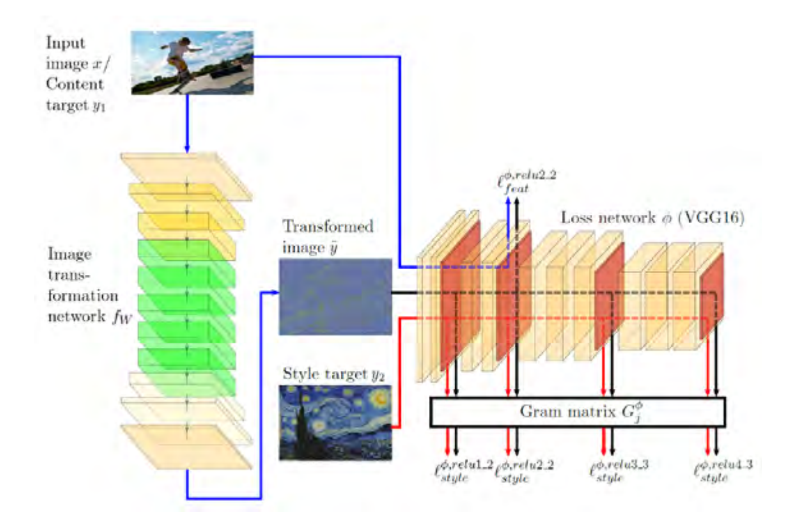


Figure 2.1: System overview. We train an image transformation network to transform input images x into output images $\hat{y}$. A loss network ϕ pre-trained for image classification is used to define loss functions. We measure the differences in content and style between the target images and the transformed image. The loss network is not changed during training.

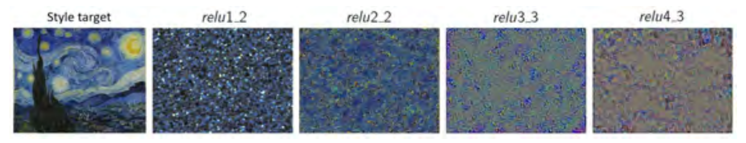
Only the output layer uses a scaled hyperbolic tangent function instead to ensure that the output image has pixels with values in the range $[0, 255]$. These encoders and decoders are connected by residual blocks.

2.2 Perceptual loss function

We apply a perceptual loss function, derived from a pre-trained network. This loss network ϕ consists of the first four blocks of the VGG-16 network [11] pre-trained on the ImageNet dataset [12]. This dataset contains a total of over 14 million human annotated images developed for computer vision research. These are organized into around 22K sub-categories, which can be considered as sub-trees of 27 higher-level categories such as animals, plants or people. The loss network is used to define a feature reconstruction loss and a style reconstruction loss, that measure differences in content and style of



(a) Images $\hat{y}$ that minimize the feature reconstruction loss $\ell_{feat}^{\phi,j}(\hat{y}, y_1)$. An image of the Microsoft COCO dataset [13] is taken as content target y_1.



(b) Images $\hat{y}$ that minimize the style reconstruction loss $\ell_{style}^{\phi,j}(\hat{y}, y_2)$. Vincent van Gogh's painting *The Starry Night* [14] is taken as style target y_2.

Figure 2.2: Reconstruction from different layers of the pretrained VGG-16 loss network ϕ. Input image is a white noise image.

the transformed image and the content and style targets, respectively. Finally, the combination of these two losses is minimized.

The feature reconstruction loss is defined as the squared, normalized Euclidean distance (mean squared error) between feature representations

$$\ell_{feat}^{\phi,j}(\hat{y}, y_1) = \frac{1}{H_j W_j C_j} \|\phi_j(\hat{y}) - \phi_j(y_1)\|_2^2, \tag{2.2}$$

where $\phi_j(x)$ is a feature map of layer j with shape $H_j \times W_j \times C_j$. In this case, H_j and W_j represent the height and width, respectively, and C_j the number of channels of the feature map. The reconstruction of images from the first layers of the loss network provides images that are perceptually similar to the target image, but that do not necessarily fit exactly, see Figure 27.2(a). We use the feature map at layer $j =$ relu2_2 of the loss network to calculate the feature reconstruction loss in Equation (2.2).

In addition to the content of the target image, the style of another image has to be met. However, the difference of two images in style is not as simple to represent as the in difference in content. Copying the feature reconstruction loss would result in comparing the content

of the style image with the output image $\hat{y}$, which is not our aim. In order to extract the style representation of the style image, only, we use the Gram matrix $G_j^\phi(x)$ to find the correlation of the channels (features) of a feature map. This approach is based on the assumption that the style of the image is defined through the co-occurrence of particular features. As in the loss function above, let $\phi_j(x)$ be the outcome of the network ϕ at layer j for the input x, which is a $H_j \times W_j \times C_j$ feature map. Then, the *Gram matrix* $G_j^\phi(x)$, which has a size of $C_j \times C_j$, is defined by

$$G_j^\phi(x)_{c,c'} = \frac{1}{H_j W_j C_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}, \tag{2.3}$$

where $c, c' \in [1, \ldots, C_j]$. Thus, we get the style reconstruction loss via

$$\ell_{style}^{\phi,j}(\hat{y}, y_2) = \| G_j^\phi(\hat{y}) - G_j^\phi(y_2) \|_F^2, \tag{2.4}$$

i. e., the squared Frobenius norm of the difference of the Gram matrices of output and target image.

Reconstruction from higher layers of the loss network transfers larger scale structure from the target image (see Figure 27.2(b)). We use this fact to reconstruct style from a set of layers J instead of a single layer j and define $\ell_{style}^{\phi,J}(\hat{y}, y)$ as the sum of losses for each layer $j \in J$. We combine the four layers *relu*1_2, *relu*2_2, *relu*3_3, and *relu*4_3 of the VGG-16 loss network ϕ for the style reconstruction loss, using all available information. Hence, we set $J = \{relu1_2, relu2_2, relu3_3, relu4_3\}$ and get the following optimization task

$$\hat{y} = \arg\min_y \left[\lambda_c \ell_{feat}^{\phi, relu2_2}(y, y_1) + \lambda_s \ell_{style}^{\phi,J}(y, y_2) \right]. \tag{2.5}$$

Here, $\lambda_c > 0$ is a content and $\lambda_s > 0$ a style weight factor. These weights have to be adjusted carefully by trial-and-error, in our case $\lambda_c = 10^5$ and $\lambda_s = 10^{10}$. To solve Equation (2.5), we use several optimization methods.

2.3 Optimization methods

We focus on comparing extensions of the stochastic gradient descent method (SGD) [15]. In SGD, the step size or learning rate in each iteration is initially selected and kept fix. The first improvement to SGD is AdaGrad, which adjusts the learning rate dynamically based on all gradients observed before. AdaDelta restricts the number of accumulated past gradients to a fixed number, instead of accumulating all past gradients. It has been developed to avoid the radical decay of learning rates observed in AdaGrad. In RProp, the idea of only using the sign of the gradient is combined with the idea of adapting the step size individually for each weight. However, the particular gradient is not available. This is improved by the use of the moving average in RMSProp, which has been developed independently of AdaGrad. It also keeps the estimates of the squared gradients, but uses a moving average instead of continually accumulating them. Finally, Adam and its variants are very popular in style transfer. Adam is similar to RMSProp and AdaDelta, but uses an exponentially decaying average of the past gradients. Compared to Adam, AdaMax scales the gradients inversely proportional to the L^∞ norm instead of the L^2 norm of the past gradients. Adam_AMSGrad maintains the maximum of all exponentially decaying averages of the gradients until the present time step and uses this maximum in place of the actual one. Nadam is a combination of Adam and Nesterov's momentum method [16].

We train on the Microsoft COCO dataset of the year 2017 [13]. It contains a total of over 123K images with annotations belonging to 80 object categories. In each step we update the weights of the image transformation network.

3 Outcome

We train the model for two epochs with default learning rate $\eta = 0.001$ and batch size $bs = 4$ recommended in [17]. During the training process, the optimal weights of the image transformation network for a style image are determined. For prediction, we pass a content image through this network and get the results for the eight considered optimization methods as shown in Figure 3.1. We take

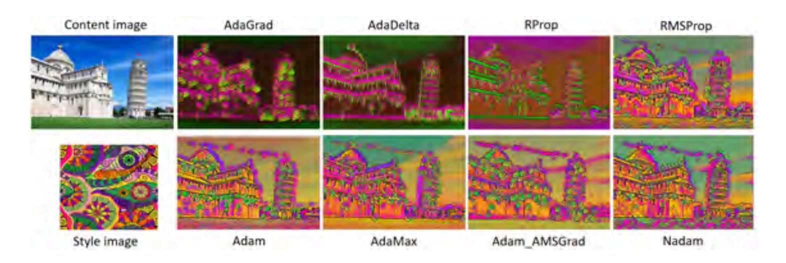
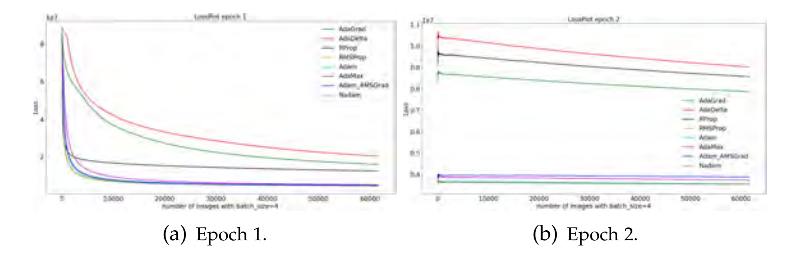


Figure 3.1: Content image (1 000 × 668), style image (800 × 800) and visualization of stylized content image with various optimizers (each 1 000 × 668).



(a) Epoch 1. (b) Epoch 2.

Figure 3.2: Loss plots for two epochs using the eight optimization methods.

an image of the Leaning Tower of Pisa [18] as content image and a colorful pattern [19] as style image. Visually, the differences are quite strong. AdaDelta, AdaGrad or RProp result in rather dark images whose content is not as well visible as in the results of the Adam versions or RMSProp. The loss plots for the eight methods over the two training epochs also clearly show the differences, see Figure 3.2. To investigate the dependence of the solution on the hyperparameters of the optimization method, we vary the learning rate $\eta \in \{0.1, 0.01, 0.001, 0.0001\}$ and batch size $bs \in \{1, 2, 4, 8\}$. The best setting and the corresponding loss values and training times are shown in Table 1. The adjusted parameters result in more similar images except for AdaGrad and RProp. The loss value for the RProp result differs by a factor of almost two from the loss values obtained by the other methods. This is also reflected in the updated results in Figure 3.3. The training times yield a similar picture: All methods

Table 1: Summary of the selected parameters, the loss values, and training times for two epochs training using the eight optimization methods.

Optimization method	η	bs	Loss value	Training time (in hours)
AdaGrad	0.01	1	$4.2293 \cdot 10^6$	5.6
AdaDelta	0.1	1	$3.3681 \cdot 10^6$	6.0
RProp	0.0001	1	$6.0191 \cdot 10^6$	8.6
RMSProp	0.001	1	$3.3217 \cdot 10^6$	5.9
Adam	0.001	1	$3.5556 \cdot 10^6$	5.8
AdaMax	0.001	1	$3.4540 \cdot 10^6$	5.8
Adam_AMSGrad	0.001	2	$3.5892 \cdot 10^6$	5.9
Nadam	0.001	2	$3.4687 \cdot 10^6$	6.0

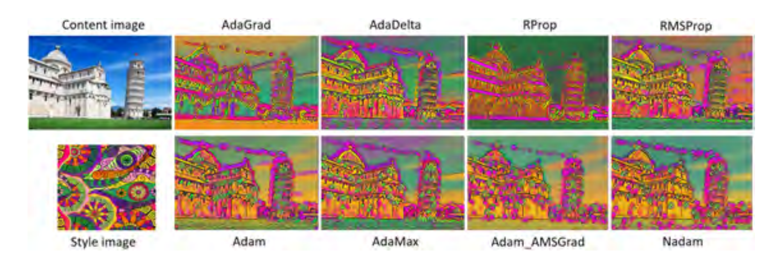


Figure 3.3: Content image (1000×668), style image (800×800) and visualization of stylized content image with different optimizer and adjusted parameters (each 1000×668).

take approximately 6 hours or less, while the training with RProp requires 8.6 hours.

Differences of the optimization methods show with respect to parameter selection, too. The Adam versions or RMSProp lead to similar loss values and stylized images, even if the selected learning rates and batch sizes are not optimal, whereas for AdaGrad, AdaDelta, and RProp the loss values depend strongly on the chosen parameters and can exceed the optimal ones by far. Comparing Figures 3.1 and 3.3 also clearly shows these differences.

4 Summary

The choice of the optimization method can be decisive for the result of the artistic style transfer. It is advisable to use one of the Adam versions or RMSProp which are more robust with respect to parameter choice than the other methods considered here. Even though we have used loss functions for measuring the quality of style reproduction, the evaluation of ”style” is rather subjective and, hence, hard to measure accurately. Thus, in the next step, we plan to investigate the effect of the optimization method on a segmentation problem where differences can be quantified more explicitly.

5 Acknowledgements

This research was supported by the Fraunhofer FLAGSHIP PROJECT ML4P.

References

1. L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” *Journal of Vision*, vol. 16, no. 12, p. 326, 2016.
2. ——, “Image style transfer using convolutional neural networks,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423, 2016.
3. J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” *Lecture Notes in Computer Science*, vol. 9906, pp. 694–711, 2016.
4. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *ArXiv*, vol. abs/1912.01703, 2019.
5. J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.
6. M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method,” *ArXiv*, vol. abs/1212.5701, 2012.

7. M. A. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: the RPROP algorithm,” *IEEE International Conference on Neural Networks*, pp. 586–591 vol.1, 1993.
8. T. Tieleman and G. Hinton, “Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude,” COURSERA: Neural Networks for Machine Learning, 2012. [Online]. Available: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
9. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.
10. T. Dozat, “Incorporating Nesterov Momentum into Adam,” *ICLR Workshop*, 2016.
11. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2015.
12. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *ArXiv*, vol. abs/1409.0575, 2014.
13. T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft COCO: Common Objects in Context,” *Lecture Notes in Computer Science*, 2014.
14. V. van Gogh, “The Starry Night,” Museum of Modern Art, New York, 1889. [Online]. Available: https://www.vangoghgallery.com/catalog/Painting/508/Starry-Night.html
15. H. Robbins and S. Monro, “A stochastic approximation method,” *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 1951.
16. A. Botev, G. Lever, and D. Barber, “Nesterov’s accelerated gradient and momentum as approximations to regularised update descent,” *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1899–1903, 2017.
17. P. Saini, “StyleTransferApp,” GitHub repository, 2019. [Online]. Available: https://github.com/puneet29/StyleTransferApp
18. P. la Quiete, “Torre pendente di Pisa,” https://poderelaquiete.it/, [accessed: September 20, 2020].
19. Colourbox, “Nahtlose bunte Muster, Stock-Vektor,” https://www.colourbox.de/vektor/nahtlose-bunte-muster-vektor-7143090, [accessed: September 20, 2020].

Hierarchical classification, counting and length measurement of fish using a stacking model approach

Raja sekar Shantha kumar, Andreas Hermann, and Daniel Stepputtis

Thünen-Institut für Ostseefischerei (OF),
Alter Hafen Süd 2, 18069 Rostock (Germany)

Abstract In this paper, the development of a hierarchical fish classification framework is presented. The conventional data collection technique for the commercial fish stock assessment is a labour intensive and time consuming procedure. The purpose of this project is to develop a framework that classifies fish species on two level semantic hierarchy label, to count the number of fishes and to measure the length of four different fish species using a small dataset. In stage 1 of the framework, the YOLOv3 convolutional neural network is used to accomplish level one semantic hierarchy label, to count the number of fishes and to measure the length of the detected fish. In stage 2, the features from the images are extracted using the VGG16 convolutional neural network. In stage 3, the stacked generalization technique is implemented to reduce the generalization error and to accomplish level two semantic hierarchy label. The classification accuracy of the stack model is 94%. The root mean square error of the fish length measurement is 1.23 cm. The accuracy in counting the number of fish depends on the detection accuracy of the stage 1 model and the classification accuracy of the stack models. Further, the results can be improved by increasing the size and diversity of the dataset.

Keywords Convolution neural network, stacked generalization, stock assessment

1 Introduction

Biological sampling is a vital procedure in marine data collection to study commercial fish stock. The conventional techniques in use include sorting the catch into species, measuring the length and counting the number of the individual catch. Since this process is labour intensive and time consuming, marine scientists are attempting to develop a deep learning framework to automate this process.

The convolutional neural network (CNN) is such an efficient deep learning technique for classifying images. A collection of tensorflow models trained using different datasets to detect common objects is given by [1]. In general, a single CNN architecture includes two parts, multiple trainable stages (feature extractor) followed by a supervised classifier (deep neural network) [2]. French et al. [3] have used CNN for detecting and counting fishes in the video footage captured on operational trawlers.

Deeper CNN's with a large number of model parameters and also trained on a huge number of examples drastically improves the classification accuracy [4]. Simonyan et al. [5] proposed a network called VGG16 in ILSVRC 2014, trained on ImageNet [6] dataset, achieves 92.7% test accuracy on the testing data. ImageNet is a dataset of nearly 15 million common object images with around 22,000 categories. ILSVRC14 uses a subset of the ImageNet dataset with 1000 images per class (1000 categories).

While there are so many fish species in the world, only a few small open source fish datasets [7] [8] are available. Practically, it is not possible to develop a generalized fish detection model using currently available datasets. To increase classification accuracy using a small dataset, Siddiqui et al. [9] used a cross-layer pooling algorithm with the CNN as feature extractor and support vector machine as a classifier to classify fish species such as P. porosus, P. emeryii and etc.

In general, a single deep learning model (feature extractor and a classifier) trained on small datasets can bias to the dataset used for the training and not performing well on unseen data (overfitting) [10]. Wolpert [11] proposed a method called stacked generalization which uses a number of base models and a single meta model to minimize the generalization error.

Human has the ability to classify a fish in a semantic hierarchy i.e Fish $\rightarrow$ Flatfish $\rightarrow$ Dab. While conventional CNN achieved remarkable performance on visual recognition, they do not recognize the object on the natural paradigm of hierarchy. Hence, there is a need in the marine field to develop a framework that allows us to classify fish species in the semantic hierarchy. Inspired by the method proposed by Wolpert [11] and combined with semantic hierarchical label classification, we propose a framework to (a) detect, (b) classify fish in the two level semantic hierarchy, (c) count the number and measure the length of fish.

2 Dataset

We used two public and one own dataset to train the models. The two public datasets are "Open images dataset" [8] and "QUT FISH dataset" [7]. The examples in the public datasets are labled with the level one label of the semantic hierarchy (Fish). The own dataset is captured in the laboratory at "Thünen-Institute (OF)" and at the fishery research vessel "Solea". Therefore, the dataset is named "Thünen dataset" and has both level one and two labels of the semantic hierarchy as shown in figure 2.1. Where the level two hierarchy refers to the fish species. Figure 2.2 show example images from "Thünen dataset".

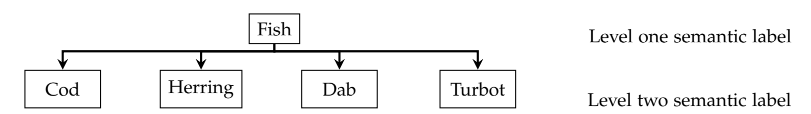


Figure 2.1: Hierarchical annotation of the dataset

Further to train the base models, "Thünen dataset" is divided into training data and testing data as shown in figure 2.3.

3 Classification Procedure

The developed framework has three stages, stage 1 – detection and classification of level one label of the semantic hierarchy, stage 2 –



Figure 2.2: (a) Cod, (b) Herring, (c) Dab, (d) Turbot

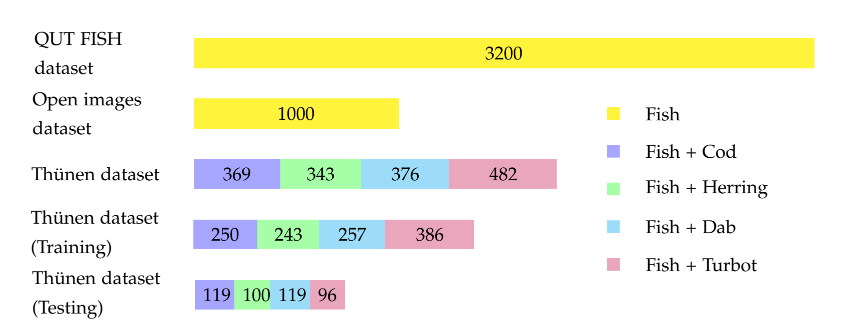


Figure 2.3: List of datasets used in training

feature extraction and stage 3 – classification of level two label of the semantic hierarchy as shown in figure 3.1. In stage 1, YOLOv3 CNN is used to detect the fish and to accomplish level one label of the semantic hierarchy. The detected fish is cropped and in stage 2, the features are extracted using VGG16 CNN. Stage 3 of the framework has a stack model with 2 layers. Layer 1 has three base models and layer 2 has a single meta model. The extracted features are used to train the three base models of the stack layer 1. Later, the prediction probabilities of the three base models are used to train the meta model of the stack layer 2. In stage 3, the level two label of the semantic hierarchy is accomplished.

3.1 YOLOv3 object detector

To detect a fish, a real time single shot object detector YOLOv3 [12] convolutional neural network is used. The YOLOv3 network is trained on the COCO dataset [13] to detect 80 common objects where

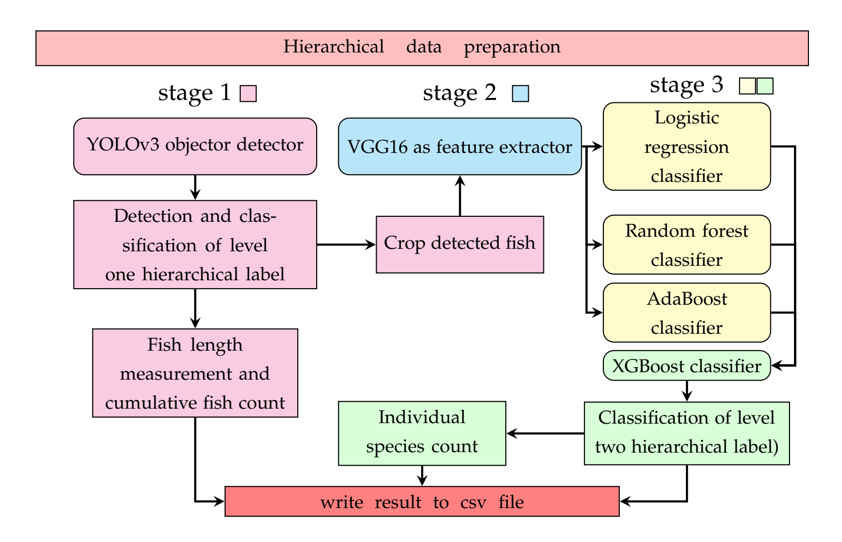


Figure 3.1: The flowchart representation of the classification procedure

fish is not one among those classes. We implemented transfer learning [14] to detect single class, Fish. Both ”QUT FISH dataset” and ”Open image dataset” with the level one label of the semantic hierarchy are used to train the model. The ”Thünen dataset” (entire dataset) with the level one label of the semantic hierarchy is used to evaluate the model performance. Figure 5.1 shows the training and validation curve of YOLOv3.

3.2 VGG16 as feature extractor

To use pre-trained VGG16 CNN [5] as a feature extractor, the last few fully connected layers were removed (modified VGG16). The image propagates from the first layer to the last layer of the modified VGG16 (feature extractor) and outputs a volume of the shape $7\times 7 \times 512$. This output volume is flattened into a feature vector of the dimensions 25,088.

To train and evaluate the base models in stage 3 (stack layer 1), the features from the ”Thünen training and testing data” are extracted and tabulated. The shape of the tabular datasets is (number of im-

ages $\times$ 25088). Figure 3.2 shows the pictorial representation of the feature map extracted from block1_conv2D layer of VGG16.

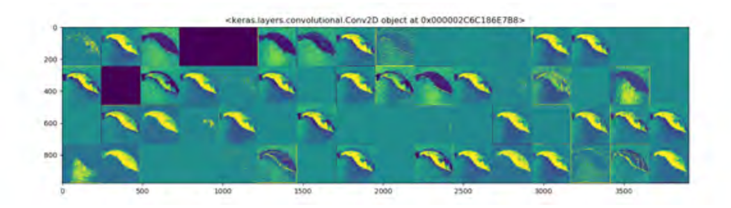


Figure 3.2: Feature map of an example image

3.3 Stacking model approach

Ensemble learning is a technique to reduce the variance of the model. Such technique for classification problems are majority voting [15], weighted majority voting [16] and stacking [11]. In majority voting, the final decision is made by a majority vote of the individual classifiers. Whereas in the weighted majority voting, the individual classifiers are assigned with different weights depending on the performance and the final decision is made by counting the weighted votes of the individual decisions [16].

The stacking or stacked generalization uses a concept meta classifier. The meta classifier is trained on the prediction probabilities of the individual base models to make the final prediction. This method reduces the generalization error and increases the prediction accuracy.

Base models

The base models used in the framework are logistic regression, random forest and AdaBoost classifier. These models are trained on the "Thünen training dataset" (features vectors) using K-fold cross validation ($K = 3$). The prediction probabilities of each base model are concatenated as shown in figure 3.3 and used as a training dataset to train the meta model.

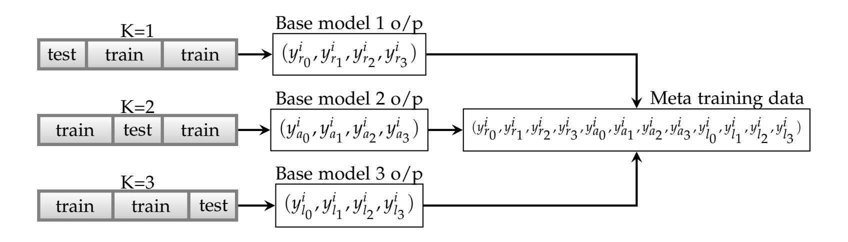


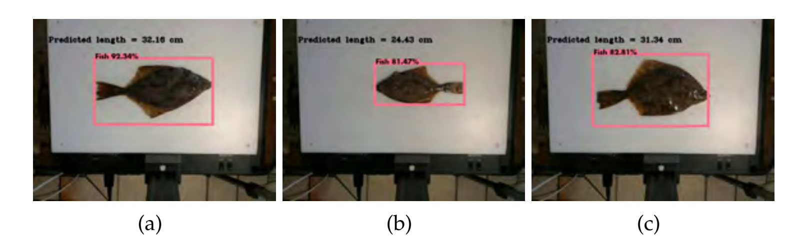
Figure 3.3: Meta training data

Meta model

The meta model used is XGBoost classifier and fitted on the prediction probabilities of the base models and the model performance is evaluated using the "Thünen testing dataset" (feature vectors).

4 Fish counting and length measurement

By using the YOLOv3 network, the overall number of fish (level one hierarchy) is counted. Similarly, the number per fish species is counted using the classification output of the stack model, following the previous detection of the YOLOv3 network.



(a) (b) (c)

Figure 4.1: (a), (b) and (c) - Predicted length using YOLOv3

The YOLOv3 network is used to predict the length of the fish. The object detection happens in the three scales and at three different layers of the YOLOv3 network, 82, 94 and 106. The input image of the shape (416, 416, 3) is downsampled by the factor (stride) 32, 16

and 8 at three detection layers and the resultant feature map has the shape of $13 \times 13 \times depth$, $26 \times 26 \times depth$ and $52 \times 52 \times depth$ respectively. For each cell in the resultant feature map, three bounding boxes are generated by the YOLOv3 network. The maximum probability of the bounding box containing a class is given by the product of objectness score and confidence. The real width b_w and the height b_h of the bounding box are computed by calculating the log-space transform (offset) to the predefined anchors. And to calculate the center coordinate (b_x, b_y) of the bounding box, a sigmoid function is used [12]. Figures 4.1 (a), (b) and (c) show three examples of the predicted length using the YOLOv3 network.

5 Results and Discussion

The training graph figure 5.1 (a) shows that the YOLOv3 network's training loss is decreasing gradually and reaches an average loss of 0.68. The mean average precision of the validation data reaches 100%.

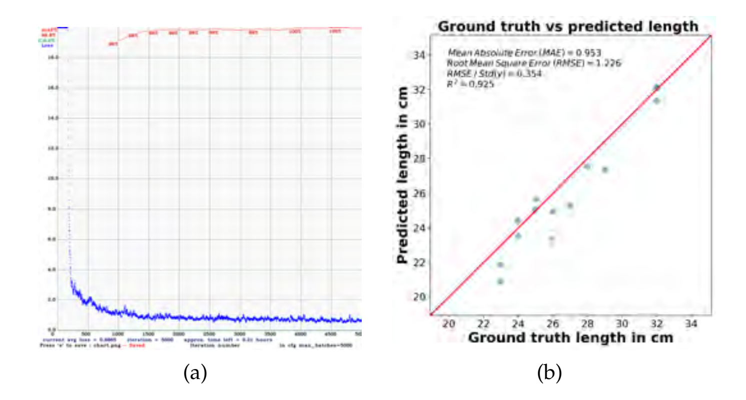


Figure 5.1: (a) YOLOv3 training curve (b) Fish length measurement plot

Figure 5.1 (b) shows the ground truth length vs predicted length of the fish plot. The root mean square error (RMSE) of the fish length measurement is 1.23 cm.

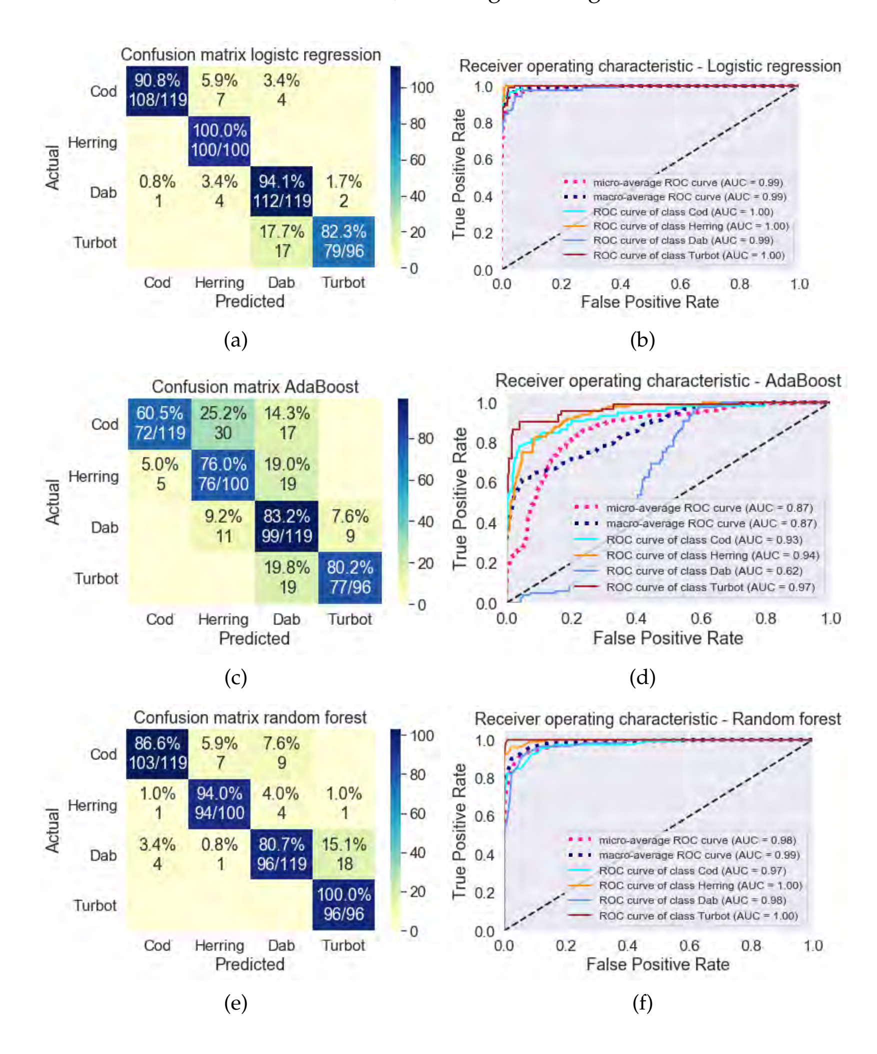
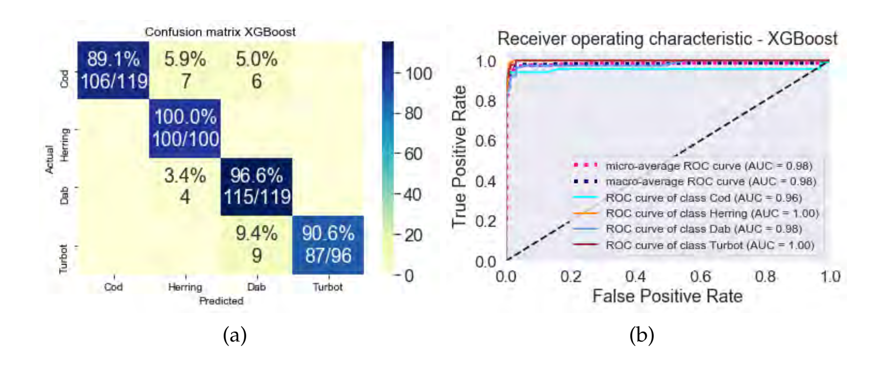


Figure 5.2: Confusion matrix of (a) Logistic regression, (c) AdaBoost and (e) Random forest. ROC curve of (b) Logistic regression, (d) AdaBoost and (f) Random forest



(a) (b)

Figure 5.3: Xgboost (a) Confusion matrix and (b) Roc

From the computed confusion matrix, figures 5.2 (a), (c), (e) and 5.3 (a) the different metrics to evaluate the stack model performance are calculated and shown in table 1. Figure 5.2 (b), (d), (f) and figure 5.3 (b) show the receiver operating characteristic curve with the area under curve value for four different classes. Comparing the classification accuracy, precision and the recall of the meta model and base models, it is clear that the meta model XGBoost out performances all three base models.

Table 1: Results of the stack models

Classifier	Precision	Recall	f1-score	Simple Accuracy	Micro AUC
Random forest	0.90	0.90	0.90	0.90	0.98
Logistic regression	0.93	0.92	0.92	0.92	0.99
Adaboost	0.78	0.75	0.75	0.75	0.87
XGBoost	0.95	0.94	0.94	0.94	0.98

6 Conclusion

From the above results, it becomes clear that the classification accuracy, precision and the recall of the fish species can be increased using a stacked generalization. The disadvantage of this approach is computationally expensive to train the model and to tune the hyper

parameter. The predicted length measurement values have relatively high root mean square error (RMSE). Therefore, the applied simple method of length estimation might not suitable for many biological applications. Hence, for further improvement, we could add more data in the training set for better accuracy of object localization or we can implement a machine vision approach such as a stereo vision for length measurement.

References

1. J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," 07 2017, pp. 3296–3297.
2. H. Lee, R. Grosse, R. Ranganath, and A. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," 01 2009, p. 77.
3. G. French, M. Fisher, M. Mackiewicz, and C. Needle, "Convolutional neural networks for counting fish in fisheries surveillance video," 09 2015.
4. D. C. Ciresan, U. Meier, L. Gambardella, and J. Schmidhuber, "Deep big simple neural nets excel on handwritten digit recognition," *ArXiv*, vol. abs/1003.0358, 2010.
5. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.
6. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.
7. K. Anantharajah, Z. Ge, C. McCool, S. Denman, C. B. Fookes, P. Corke, D. W. Tjondronegoro, and S. Sridharan, "Local inter-session variability modelling for object classification," in *Winter Conference on Applications of Computer Vision (WACV), 2013 IEEE Conference on*, 2014.
8. I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, S. Kamali, M. Malloci, J. Pont-Tuset, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy, "Openimages: A public dataset for large-scale multi-label and multi-class image classification." *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.

9. S. Siddiqui, I. Malik, F. Shafait, A. Mian, M. Shortis, and E. Harvey, "Automatic fish species classification in underwater videos: Exploiting pretrained deep neural network models to compensate for limited labelled data," *ICES Journal of Marine Science*, vol. 75, 05 2017.
10. X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, p. 022022, feb 2019. [Online]. Available: https://doi.org/10.1088%2F1742-6596%2F1168%2F2%2F022022
11. D. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 12 1992.
12. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 04 2018.
13. T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014.
14. K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, May 2016. [Online]. Available: https://doi.org/10.1186/s40537-016-0043-6
15. L. Lam and S. Y. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 27, no. 5, pp. 553–568, 1997.
16. A. F. R. Rahman, H. Alam, and M. C. Fairhurst, "Multiple classifier combination for character recognition: Revisiting the majority voting system and its variations," in *Document Analysis Systems V*, D. Lopresti, J. Hu, and R. Kashi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 167–178.

Binary Maps for Image Separation in Iterative Neuronal Network Applications

Roman Lehmann, Stanislav Arnaudov, Markus Hoffmann, and Wolfgang Karl

Karlsruhe Institute of Technology, Institute of Computer Engineering,
Kaiserstraße 12, 76131 Karlsruhe

Abstract Generating a series of images is an important task in various fields of scientific research, e. g. Computational fluid dynamics (CFD). In the past years, solutions based on deep neural networks gained importance. In these tasks, it's often necessary to declare regions of interest in the image. Furthermore, the NN should only perform on these regions and the rest should be ignored. With this paper, we propose an innovative and easy method for implementing this behavior in the field of CFD.

Keywords U-net, binary maps, generator, flow simulation

1 Introduction

Generating a series of images is an important task in various fields of scientific research, e. g. Computational fluid dynamics (CFD). In the past years, solutions based on deep neural networks gained importance [1], for example in applications where the results don't need to be fully accurate. For these tasks, it's often necessary to declare regions of interest (ROI) in the image to preserve constant regions and concentrate the influence of the neuronal network on a specific area. This becomes even more important in cases where results of a neuronal network are used as an input again. We call these cases iterative applications.

For CFD applications this issue is related to the sharp separation of the simulation area and its boundary. It is essential that the fluid simulation does not ignore boundaries, like obstacles within the stream,

and that these boundaries do not introduce false interferences into the simulated stream. Using an image-to-image approach [2] to create a sequence of simulation steps, an obvious idea to define a sharp separation is the usage of binary maps. Such maps define a region of interest within a picture with a true value for the corresponding pixel and false otherwise. In combination with a neuronal network, these maps can be used as an additional parameter track for the network and as a filter for a post image processing step. We will show that both applications are needed in order to get good predictions for the simulation results with a sharp separation of the simulation area and its boundary.

2 State of the art

The main task in our approach is an image-to-image translation. By now the image-to-image translation through CNNs is well established and has found numerous applications [3–6]. [2] has specifically stated how the ″community-driven research″ has popularized their work by applying it in different ways [7–9]. We see our work as another demonstration of [2]. This time in the context of CFD.

The field of CNNs provides various approaches of handling with ROIs. [10, 11] use different NNs for generating and applying the ROIs. This leads to results with a probability, which is desired in the given tasks, but not in ours. Other works like [12, 13] use binary mask to define ROIs. But they use these mask as an pre image processing step only. Our application of the binary map goes further with respect to the combined application.

3 Methodology

The task is to build a network that can predict the next frame of a 2D flow simulation based on the previous one. Our focus of this work is on the boundaries of the simulation area, obstacles for example, and their stability in iterative evaluations of the network. Each frame represents a time step of the simulation and consists of a three-channel image. Two of the channels encode the velocity fields in x- and y-direction and the third channel is the pressure field of the fluid. For

this paper we did not construct a single holistic model that can handle all the simulation's parameters. Our effort is concentrated on taking a relative simple model and investigate the influence of the application of binary maps. We call this simple model the *constant model* because we do not vary simulation parameters like the inflow speed.

3.1 Simulation setup and data generation

The training data was generated by performing simulations of incompressible fluid flow around a rectangular object in a channel. The simulations are modelled according to the Navier-Stokes equations for in-compressible flow. Because we are interested in the image representations of the simulations, we are dealing only with the 2D case. Several boundary conditions describe the simulation setup:

- Inflow condition on the left side of the channel
- Outflow condition on the right side of the channel
- No-slip condition on the bottom and top side of the channel as well as the sides of the object.

The simulation setup has three separate adjustable parameters: inflow speed g, fluid density ρ and fluid kinematic viscosity v. For the constant model we took the simulation with $\rho = 0.2$, $v = 0.0009$ and $g = 1.5$. The choice of the parameter is deliberate. The values are chosen so that the Reynolds number [14] of the simulations in the range of $[90, 450]$. We were interested whether the build models can predict the emerging *Kármán vortex street* [15]. Thus, the Reynolds numbers were chosen so that the effect can occur.

The simulations were performed numerically by solving the differential equation describing the flow – the Navier-Stokes equations. This was done with a numerical solver library – *HiFlow*[3] [16] – that works on the base of the finite element method [17]. The time step for the solver was set to 0.035 seconds. This means a single time step of the simulation corresponds to 0.035 seconds of physical time.

The numerical solver library on itself cannot be used to render the simulation results to images. For this reason, we used *ParaView* [18]

to load the simulation data and exported it as a sequence of images in *PNG* format. We used the default "Grayscale" color preset of ParaView to visualize the results. Each frame of the simulation was exported as three separate grayscale images. Finally, the images were cropped to select a subset of the space that contains the object and space behind it. For training the neuronal network, we rendered 1904 frames of the simulation (66 seconds of the simulated physical time).

After all images were generated, a test-train split was created. The split was done by random and resulted in 80% of the data was used for training and the rest for testing.

The binary map was created be locating the obstacle and set the size to the same length and width like the other images.

3.2 Training approach and network details

We based our generative models almost entirely on [2]. We use the conditional GAN approach to train a generator network that can perform image-to-image translation. As explained in [2], the traditional GAN method uses a random vector z as an input to the generator network G to generate output y, $G : z \to y$. Conditional GANs also feed an input image x to the generator, $G : x, z \to y$. [2] and [19] suggest that in certain cases the usage of z can be usefully, but we decided not to include for our generator as we want a deterministic network. The discriminator network is modelled with the function $D : x, y \to v$ that evaluates the likelihood of y being a real image. To note is that the discriminator network has access to the real image x and tries to guess, if y is the real or generated output.

We adopt the objective function of the discriminator network and we modify it slightly by leaving out the random vector z.

$$\begin{aligned} \mathcal{L}_{cGAN}(G, D) &= (\mathbb{E}[\log D(x, y)] + \mathbb{E}[\log D(x, G(x))])/2 \\ &= (\log D(x, y) + \log D(x, G(x)))/2 \end{aligned} \tag{3.1}$$

where x is the input image and y is the target image. We leave out the expected value calculation as we do not use the random vector z in our loss function. In contrast to unconditional GANs, both the generator and the discriminator network have access to the input

image. The objective is divided by two to slow down the training of the discriminator relative to the generator as suggested by [2].

The objective for the generator network is composed of two parts — the value of the discriminator as well as a $L1$ distance loss between the target and the predicted images. According to [2] the $L1$ loss promotes less blurring and captures the low frequency details of the images. The $L1$ loss is given by:

$$\mathcal{L}_{L1}(G) = \mathbb{E}[\|y - G(x)\|_1] \tag{3.2}$$

The final object for the generator is thus:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN} + \lambda \mathcal{L}_{L1}(G) \tag{3.3}$$

For all models we used $\lambda = 100$ as done in [2].

3.3 Network architecture

For our generator we use the U-Net [20] variant proposed in [2]. It is a standard encoder-decoder [21] model with skip connections between parts of the encoder and the decoder. Our network uses blocks of layers of the form convolution-normalization-ReLu [22]. The encoder-decoder first downsamples the input till a bottleneck layer is reached and what follows is an upsampling to the original size of the input image.

For the discriminator, we follow the method of [2] and we use their PatchGAN discriminator network. This is a convolutional network that classifies patches of the input as real or predicted. To note is that the whole image is given as an input. The majority of the results in [2] show that patches of size 70×70 yield the best results but in our case, the experiments showed otherwise. We, therefore, we opted out for using patches of size 286×286 pixels.

3.4 Training details

We trained the model with the generated dataset. When loading the images in memory, we first resize them to an appropriated for a network size of 1024×256 (width $\times$ height). Then we apply random

crops as well as add random noise to each channel of the images. We do this to force the generator to learn the actual features of the simulation and make over-fitting harder. To investigate the effect of using a binary map to determine the obstacle. We developed four different training:

- **no-mask:** no binary mask is used at all
- **no-mask-after:** the binary mask is multiplied to the input image. The binary mask itself is also fed as additional input into the generator network but not multiplied with the predicted image.
- **no-mask-before:** only the predicted image is multiplied with the binary image. No binary mask is fed into the network or is multiplied to the input image.
- **mask:** the binary mask is multiplied to the input image. The binary mask itself is also fed as additional input to the generator network. The predicted image is multiplied with the binary image, too.

The binary map as additional input gives the network the information where the obstacle is. The zeroed values can't provide this information due the grayscaled image.

For the training procedure, we follow the standard approach in [23]. With each mini-batch, we first optimize the discriminator and then the generator with the discussed objectives. We use Stochastic Gradient Descent [24] with the Adam optimizer [25] with a learning rate of 0.0002 and standard momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The used batch size for the constant model was set to 3. Those are relatively small numbers for batch sizes but [2] suggests that the U-Net architecture benefits from small batches in image-to-image translation problems.

The constant model was trained for 45 epochs and evaluated on a single Nvidia GTX 980Ti GPU. The Implementation of the models was done in *PyTorch* [26] python library for machine learning.

4 Results

At this point we want to mention why the following results are showing the beginning of the vortex street and not a fully distinctive turbulent flow. The reason can be found in the training data and the very short amount of time the vortex street needs to establish within the stream. Therefore, the neuronal network is well-trained to predict the continuation of the distinctive turbulent flow but less highly trained for the first simulation steps where the vortex street is establishing. That is why prediction problems have a higher impact on the first steps and are therefore more visible in these images, although the same problems can be observed in all simulation steps as shown later on.

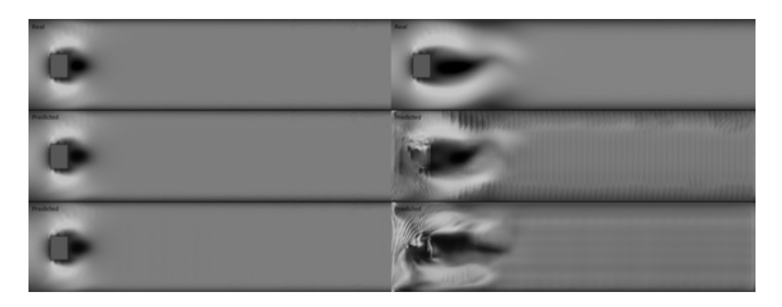


Figure 4.1: 1. Line: Step 1 and step 20 of a finite element simulation, 2. Line: Predicted step 1 and step 20 without the usage of the pressure field, 3. Line: Predicted step 1 and step 20 with usage of the pressure field; No binary mask used, x-velocity shown

We start with the mask-free prediction. Figure 4.1 shows what happens: The obstacle vanishes within the stream and this has in return a bad impact on the stream itself. Even adding more information by using the pressure field of the stream in addition to the velocity field for prediction isn't a solution.

As the first step prediction seams to be useful, the intuitive next development is to multiply every prediction with the binary mask before using it iteratively as the new input data. Results for that are shown in figure 4.2. One can see, that this idea also leads to insuffi-



Figure 4.2: 1. Line: Step 1 and step 20 of a finite element simulation, 2. Line: Predicted step 1 and step 20 without the usage of the pressure field, 3. Line: Predicted step 1 and step 20 with usage of the pressure field; Binary mask used after prediction, x-velocity shown

cient results. Adding more information with the pressure field even produces worse results with respect to the accuracy of the stream.

After observing that the simple post image processing step isn't the solution, we turned it the other way round and set the binary map as an additional data stream for the neuronal network. The idea here is that the network is able to learn the sharp separation with the help of this map. In figure 4.3 it is obvious that this isn't the right way either.



Figure 4.3: 1. Line: Step 1 and step 20 of a finite element simulation, 2. Line: Predicted step 1 and step 20 without the usage of the pressure field, 3. Line: Predicted step 1 and step 20 with usage of the pressure field; Binary mask used within neuronal network, x-velocity shown

Combining both approaches, adding the binary map to the neuronal network and using it for post-processing the result, is the next logical step at this point. Figure 4.4 shows, that this approach preserves the obstacle perfectly and results in good predictions. There are relics on the image, but they are very homogeneous and can be filtered with common image processing steps like opening and closing. The stream itself is in both predictions very close to the numerical simulation.

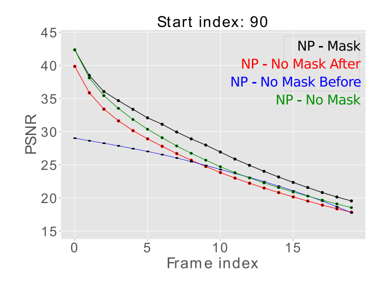


Figure 4.4: 1. Line: Step 1 and step 20 of a finite element simulation, 2. Line: Predicted step 1 and step 20 without the usage of the pressure field, 3. Line: Predicted step 1 and step 20 with usage of the pressure field; Binary mask used combined, x-velocity shown

For a real quality quantification we used a measurement to compare different images with the focus on the human observer. Implying that the result doesn't have to be fully accurate, we used the Peak Signal Noise Ratio (PSNR) as the metric. It is connected to the mean square error (MSE) in the following way:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{255}{\mathrm{MSE}}\right) \quad \text{[decibel]}. \tag{4.1}$$

Higher values are connected to less observable differences, in general a PSNR over 30 means that the human eye cannot detect any difference [27, 28]. We started the PSNR evaluation at simulation step 90 to show that even in the well-trained time steps of the simulation where the vortex street is completely visible a relevant difference is measurable. Figure 4.5 not only shows that the combined approach results in the best predictions but also that a bad application of the



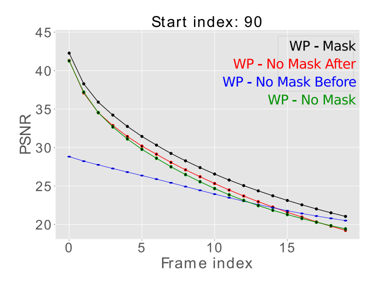


Figure 4.5: Left: PSNR values for 20 iterative steps, starting with step 90, no pressure field used; Right: Added pressure field.

binary map can result in even worse predictions than applying no binary map.

5 Summary

Defining regions of interest with the help of binary masks for iterative neuronal network applications like predicting CFD results is an important issue for such predictions. As seen in figure 4.1 to 4.4 applying no binary mask leads to wrong results very quickly. Applying only one approach — train the mask or using it as a post-processing step — can preserve the obstacle but cannot avoid interferences on the stream. Only applying both strategies results in appropriate predictions even when more information, like the pressure field, is used. The PSNR values in figure 4.5 are showing that this is even true for very well-trained parts of the simulation. This figure also demonstrates that a wrong application of a binary mask can lead to worse predictions than applying no mask at all. Therefore, we suggest a combined application of a binary mask for iterative network applications with sharp separations of regions of interest.

References

1. O. Hennigh, “Lat-net: Compressing lattice boltzmann flow simulations using deep neural networks,” 2017.

2. P. Isola, J. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *CoRR*, vol. abs/1611.07004, 2016. [Online]. Available: http://arxiv.org/abs/1611.07004
3. B. Zhao, W. Yin, L. Meng, and L. Sigal, “Layout2image: Image generation from layout,” *International Journal of Computer Vision*, pp. 1 – 18, 2020.
4. Y. Liu, Z. Qin, Z. Luo, and H. Wang, “Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks,” *CoRR*, vol. abs/1705.01908, 2017. [Online]. Available: http://arxiv.org/abs/1705.01908
5. M. D. Zeiler, G. W. Taylor, and R. Fergus, “Adaptive deconvolutional networks for mid and high level feature learning,” in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2018–2025.
6. T. Park, M. Liu, T. Wang, and J. Zhu, “Semantic image synthesis with spatially-adaptive normalization,” *CoRR*, vol. abs/1903.07291, 2019. [Online]. Available: http://arxiv.org/abs/1903.07291
7. S. Moschoglou, S. Ploumpis, M. Nicolaou, A. Papaioannou, and S. Zafeiriou, “3dfacegan: Adversarial nets for 3d face representation, generation, and translation,” *ArXiv*, vol. abs/1905.00307, 2019.
8. B.-K. Kim, G. Kim, and S.-Y. Lee, “Style-controlled synthesis of clothing segments for fashion image manipulation,” *IEEE Transactions on Multimedia*, vol. 22, pp. 298–310, 2020.
9. S. S.-C. Chen, H. Cui, M. Du, T. Fu, X. S. Sun, Y. J. Ji, and H. Duh, “Cantonese porcelain classification and image synthesis by ensemble learning and generative adversarial network,” *Frontiers of Information Technology & Electronic Engineering*, vol. 20, pp. 1632 – 1643, 2019.
10. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International conference on machine learning*, 2015, pp. 2048–2057.
11. L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, “Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5659–5667.
12. J. Dai, K. He, and J. Sun, “Convolutional feature masking for joint object and stuff segmentation,” *CoRR*, vol. abs/1412.1283, 2014. [Online]. Available: http://arxiv.org/abs/1412.1283

13. S. Eppel, "Setting an attention region for convolutional neural networks using region selective features, for recognition of materials within glass vessels," *arXiv preprint arXiv:1708.08711*, 2017.
14. K. T. Trinh, "On the critical reynolds number for transition from laminar to turbulent flow," 2010.
15. T. v. Kármán, "Ueber den mechanismus des widerstandes, den ein bewegter körper in einer flüssigkeit erfährt," *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, vol. 1911, pp. 509–517, 1911. [Online]. Available: http://eudml.org/doc/58812
16. S. Gawlok, P. Gerstner, S. Haupt, V. Heuveline, J. Kratzke, P. Lösel, K. Mang, M. Schmidtobreick, N. Schoch, N. Schween, J. Schwegler, C. Song, and M. Wlotzka, "Hiflow3 – technical report on release 2.0," *Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)*, vol. 0, no. 06, 2017. [Online]. Available: https://journals.ub.uni-heidelberg.de/index.php/emcl-pp/article/view/42879
17. G. Strang and G. Fix, *An Analysis of the Finite Element Method*. Wellesley-Cambridge Press, 2008. [Online]. Available: https://books.google.de/books?id=K5MAOwAACAAJ
18. J. Ahrens, B. Geveci, and C. Law, "Paraview : An end-user tool for large data visualization," *Energy*, vol. 836, p. 717–732, 2005. [Online]. Available: http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:ParaView:+An+end-user+tool+for+large+data+visualization$\#$0
19. X. Wang and A. Gupta, "Generative image modeling using style and structure adversarial networks," 2016.
20. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597
21. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. [Online]. Available: https://science.sciencemag.org/content/313/5786/504
22. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

23. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

24. J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952. [Online]. Available: http://www.jstor.org/stable/2236690

25. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

26. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

27. D. Mehra, "Estimation of the image quality under different distortions," *International Journal Of Engineering And Computer Science 8*, 2016.

28. Y. Shiao, T. Chen, K. Chuang, C. Lin, and C. Chuang, "Quality of compressed medical images," *Journal of digital imaging : the official journal of the Society for Computer Applications in Radiology 20*, 2007.

Weizenährenerkennung mithilfe neuronaler Netze und synthetisch generierter Trainingsdaten

Lukas Lucks[1], Laura Haraké[1] und Lasse Klingbeil[2]

[1] Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung, Gutleuthausstraße 1, 76275 Ettlingen
[2] Universität Bonn, Institut für Geodäsie und Geoinformation, Nußallee 17, 53115 Bonn

Zusammenfassung This paper investigates the usability of synthesized training data for the recognition of wheat ears using neural networks in the context of semantic image segmentation. For this purpose, detailed scenes of wheat fields consisting of 3D models with high-resolution textures and defined material properties are modeled. Afterwards, photo realistic color images are synthesized, which also contain a binary image mask with the locations of the ear models. The resulting image pairs are then used as a training data for two neural networks (U-Net and DeepLab-V3+). To determine whether these data allows domain adaptation, the trained networks are evaluated using real wheat field images. The IoU value of about 69.96 shows that information transfer from the synthesized images to real images is possible.

Keywords Semantic segmentation, synthetic data, photorealistic rendering, domain adaptation

1 Einleitung

Um die Nahrungssicherheit für die wachsende Weltbevölkerung sicherzustellen, werden immer höhere Anforderungen an die landwirtschaftliche Produktion gestellt. Die Erfüllung der Anforderungen wird durch die weltweit steigende Flächenkonkurrenz erschwert. Durch diese Entwicklungen ergibt sich die Notwendigkeit,

die vorhandenen Flächen nachhaltig zu bewirtschaften und Pflanzensorten zu züchten, die eine effizientere Produktion ermöglichen. In diesem Kontext nimmt Weizen, als eine der wichtigsten Kulturpflanzen neben Mais und Reis, eine besondere Rolle ein. Um den Weizenanbau in Zukunft nachhaltig und effizient zu gestalten und eine präzisere Bewirtschaftung zu ermöglichen, ist eine ständige Analyse des Pflanzenwachstums notwendig. Je nach Wachstumsphase der Pflanze sind tägliche Erfassungungen erforderlich, welche wiederum durch die oftmals manuelle Durchführung sehr zeitaufwendig sind [1]. Von besonderem Interesse ist dabei die Erkennung der Ähren, da sich aus diesen relevante Bestandsparameter wie die Pflanzendichte oder das Reifestadium der Pflanzen bestimmen lassen.

Unter Verwendung von Kamerabildern und moderner Bildverarbeitungsalgorithmen wird versucht, diese Informationen automatisiert abzuleiten [2]. Diese Algorithmen lernen dabei mithilfe von Referenzdaten, die Ähren innerhalb der Bilder zu erkennen. Um das jeweilige domänenspezifische Wissen aus diesen Daten auf bisher unbekannte Bilder zu übertragen, ist eine große Menge an annotierten Beispielen notwendig. Diese müssen meist manuell und somit sehr zeitintensiv erstellt werden. Eine Möglichkeit diesen Aufwand zu minimieren, besteht darin, auf reale Daten zu verzichten und diese durch synthetisch erzeugte Bilder zu ersetzen. Eine synthetische Umgebung ermöglicht dabei eine einfache Modifizierung und effiziente Reproduktion der Daten sowie die schnelle Erstellung exakter Annotationen.

In diesem Paper wird untersucht, inwiefern das in den synthetisch erzeugten Bildern enthaltene Wissen mittels neuronaler Netze auf reale Bildaufnahmen adaptiert werden kann. Auf Basis quasi-prozedural erzeugter Weizenmodelle werden realitätsnahe Bilder eines virtuellen Weizenfeldes erzeugt. Diese dienen als Trainingsgrundlage für eine semantische Segmentierung, wobei unterschiedliche Netzarchitekturen verwendet werden (s. Kapitel 3). Die Übertragbarkeit der Ergebnisse auf reale Daten wird anhand realer Bildaufnahmen evaluiert (s. Kapitel 4). Der Ablauf des Verfahrens ist in Abbildung 1.1 zu finden. Ein Überblick über den Stand der Forschung in diesem Themenbereich wird im Kapitel 2 gegeben.

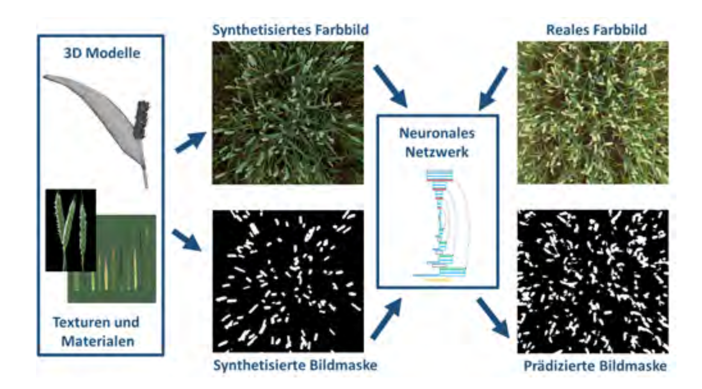


Abbildung 1.1: Übersicht der Ährendetektion. Auf Grundlage von 3D Modellen werden synthetische Farbbilder und Bildmasken erstellt. Mithilfe dieser wird ein neuronales Netz trainiert, welches die Prädiktion realer Bilder ermöglicht.

2 Stand der Forschung

Methodisch lassen sich bei der **Erkennung von Weizenähren** Deep Learning Ansätze von merkmalsbasierten Verfahren unterscheiden. Bei letzteren werden verschiedene Farb-, Textur- [1] oder Kantenmerkmale [3] definiert, welche eine pixelweise Detektion der Ähren innerhalb der Bilder ermöglichen. Dabei werden schwellwertbasierte Klassifikatoren sowie klassische Klassifikations- oder Clusterverfahren verwendet. Neuere Methoden dagegen basieren häufig auf Convolutional Neural Networks (CNNs) (s. [4] oder [5]). Weiterhin ermöglicht DeepCount [6] die Erkennung der Ähren, indem basierend auf Superpixeln diverse Merkmale berechnet und mittels eines CNNs analysiert werden. Bei [7] werden Farbinformationen mit thermalen Informationen verknüpft, um die Ähren zu identifizieren. Bei [8] wird ein semi-überwachtes Verfahren vorgestellt. Um den Annotationsaufwand für die Datengrundlage des Netzwerkes zu minimieren, wird die Idee des Aktiven Lernens auf das Deep Learning übertragen.

Die Nutzung von photorealistischen **synthetischen Datensätzen** für die semantische Segmentierung im Kontext von Computer Vision Anwendungen wird in [9] evaluiert. Die Grundlage bildet eine

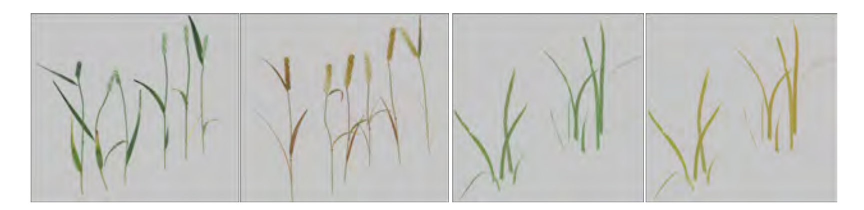


Abbildung 3.1: Verwendete Weizenpflanzen- (links) und Grasmodelle (rechts) verschiedener Reifestadien. Während die Weizenmodelle in Farbe, Textur, Länge der Blätter, Ähren und in Ausprägung der Grannen variieren, sind bei den Grasmodellen lediglich die Texturen an den Reifegrad angepasst.

prozedural generierte, komplexe Szene, deren Geometrien aus der jeweiligen Perspektive physikalisch-basiert gerendert werden. Für jedes Trainingsbild wird dabei die Szene durch die von dem Benutzer definierten Parameter neu instanziiert. Andere Verfahren wie ProcSy [10] nutzen eine prozedurale Modellierungssoftware wie CityEngine® in Kombination mit Gaming-Engines, um photorealistischen Trainingsdaten zu erzeugen.

Domain Randomization beschreibt die Idee, die Verteilung der gerenderten Daten so zu variieren, dass das neuronale Netz, welches mit diesen Daten trainiert wird, robust genug ist, auch auf den realen Daten zu funktionieren. Dabei können sowohl Position, Ausrichtung oder Materialeigenschaften der zu synthetisierenden Inhalte variiert werden. Insbesondere spielen auch Beleuchtungseigenschaften, wie Intensität und Ausrichtung von Lichtquellen, als auch Renderingparameter eine große Rolle [11].

In dieser Arbeit wird eine fixe Szene mit manuell aufbereiteten Modellen verwendet, deren Diversität über Randomisierung weniger Parameter und einen virtuellen Kameraflug erreicht werden kann.

3 Methoden

Bei der **Bildsynthese von Weizenpflanzen** wird auf die freie Software Blender® zurückgegriffen. Mit dieser lässt sich eine beliebig große Szene aus nur wenigen 3D Modellen quasi-prozedural zusammenstellen, ohne jedes einzelne Szenenobjekt bei Anpassungswünschen

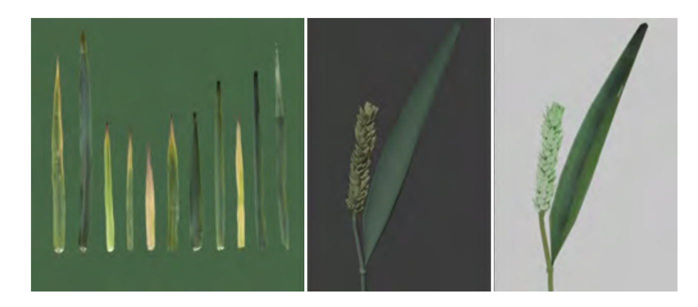


Abbildung 3.2: Manuell aufgenommene Texturen für ein frühes Reifestadium von Weizenmodellen (links), Modell einer Weizenpflanze versehen mit Materialeigenschaften (mittig), das gleiche Pflanzmodell texturiert und physikalisch korrekt gerendert (rechts).

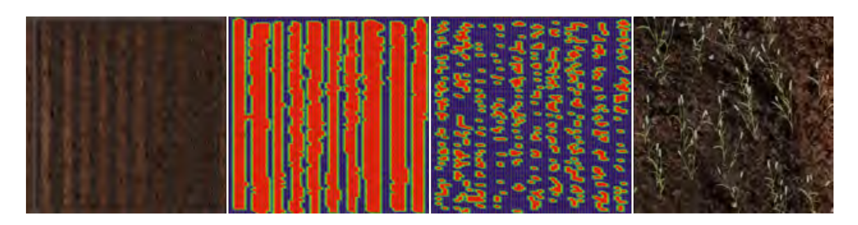


Abbildung 3.3: Von links nach rechts: Bodentextur für Weizenfeld, Platzierungskarte für Grasmodelle, Positionen für ein Weizenpflanzenmodell randomisiert auf dem Feld, Ausschnitt der Positionierung des ersten Weizenpflanzenmodells auf dem Feld.

von Geometrie- oder Materialeigenschaften bearbeiten zu müssen. Zusätzlich lässt sich die Szene unter variablen Aufnahmepositionen und Beleuchtungssituationen physikalisch korrekt rendern, sodass sich Bilder in großer Menge erstellen und beliebig reproduzieren lassen. Zu jeder photorealistischen Aufnahme wird ein Maskenbild aus gleicher Aufnahmerichtung und -höhe erzeugt. Somit ist es möglich automatisiert große Mengen an annotierten Beispieldaten zu erstellen, die für eine Bildsegmentierung genutzt werden können.

Grundlage für die **Modellierung des virtuellen Weizenfeldes** sind jeweils sechs variierende 3D Modelle von Weizenpflanzen und zwei unterschiedliche Modellgruppen von Grashalmen. Alle werden entsprechend eines Ziel-Reifegrades manuell angepasst und mit rea-

listischen Oberflächeneigenschaften versehen (Abb. 3.1). Dabei bilden Aufnahmen von realen Pflanzenblättern und Grashalmen die RGB-Textur für die Blattmodelle (Abb. 3.2). Die Materialeigenschaften der Weizenähren und -stängel sind über einen optischen Vergleich mit Aufnahmen von realen Pflanzen im jeweiligen Reifestadium festgelegt. Für die Grasmodelle werden frei verfügbare Texturen verwendet, die entsprechend angepasst sind.

Die Modellierung der Weizen- und Grasmodelle im virtuellen Weizenfeld lässt sich als quasi-prozedural beschreiben, da die Zufälligkeit der Farben und Texturen durch manuelle Auswahl beschränkt wird. Gleichzeitig kann jedoch durch die Anordnung der Pflanzen selbst eine optisch ausreichende Variabilität erreicht werden. Von jedem der sechs Weizenmodelle werden je nach gewünschter Dichte mindestens 3000 Instanzen zufällig auf dem gesamten Feld verteilt (Abb. 3.3). Dabei werden auch Höhe und Ausrichtung jeder Instanz in einem gewissen Intervall randomisiert. Die zusätzliche Verwendung der Grasmodelle trägt zu einer realitätsnahen Abbildung der Szene bei. Beispiele der synthetisierten Bilder des virtuellen Weizenfeldes sind in Kapitel 4 dargestellt und bewertet.

Das Ziel dieser Arbeit ist die **semantische Bildsegmentierung zur Ährenerkennung**, d. h. jeder Bildpixel soll dabei entweder als Ähre oder Hintergrund klassifiziert werden. Das hierfür notwendige Wissen soll mithilfe eines neuronalen Netzes aus den erzeugten synthetischen Bildpaaren adaptiert werden. Für diese Aufgabe wird sowohl das U-Net [12] als auch das DeepLab-V3+ [13] verwendet. Die Layer der Netze weisen eine klassische Encoder-Decoder-Struktur auf. Innerhalb des Encoders werden die Informationen des Eingangsbildes sukzessive verdichtet, sodass eine semantische Interpretation ermöglicht wird. Die räumliche Auflösung der einzelnen Layer nimmt mit jeder Verdichtung ab. Die durch den Encoder verlorene räumliche Information, wird durch den Decoder wiederhergestellt, sodass eine pixelweise Segmentierung des Eingangsbildes ermöglicht wird.

Der Encoder des U-Nets besitzt eine klassische Kaskade von Convolutional und Pooling Layern, deren Struktur gespiegelt im Decoder wiederzufinden ist. Dagegen besteht der Encoder des DeepLabs aus Atrous Convolutional Layern. Diese ermöglichen es, Fea-

tures mit hoher Kontextinformation zu berechnen, ohne dass die räumliche Auflösung der einzelnen Layer zu stark reduziert wird und somit ein schärferes Segmentierungsergebnis erzielt werden kann. Der Decoder des Netzwerkes besteht aus einfachen Upsampling und Convolutional Layern, die zusammen mit einigen Low-Level Features des Encoders das finale Ergebnis liefern. Als Kostenfunktion wird die binäre Kreuzentropie zum Trainieren der Netze verwendet.

4 Ergebnisse und Diskussion

Im folgenden Abschnitt werden die Ergebnisse der Bildsynthese und ihre Eignung als Trainingsdaten zur Ährendetektion erörtert.

Synthetisch generierte Trainingsdaten zur Ährenerkennung

In Abbildung 4.1 sind zwei Ergebnisse des physikalisch-basierten Renderers von Blender® dargestellt. Die gerenderten Bilder vermitteln einen photorealistischen Eindruck der Szene. Die Verteilung, Farbe, Größe und Position der Elemente sind vergleichbar mit deren Ausprägung in realen Aufnahmen. Ein erkennbarer Unterschied lässt sich allerdings bei der Darstellung des Untergrundes feststellen, da die verwendete Textur starke Glanzlichter aufweist, welche nicht gesondert aufbereitet wurden. So wirken die künstlichen Bilder an diesen Stellen dunkler als in der Realität.

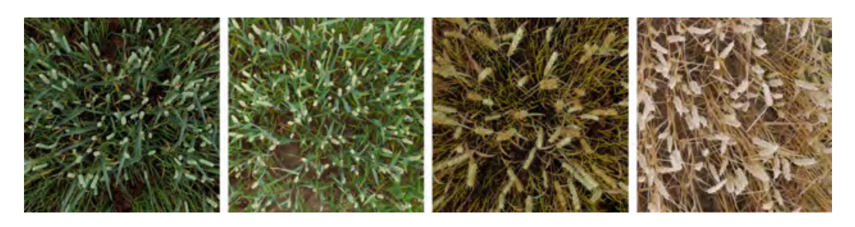


Abbildung 4.1: Ergebnis der Bildsynthese. Von links nach rechts ist jeweils ein synthetisches Bild und eine vergleichbare reale Aufnahme dargestellt.

Zu jedem synthetischen Farbbild ist eine passende Bildmaske der Ähren vorhanden. Auf Basis dieser Bildpaare werden zwei Trainings-

datensätze $T_{grün}$ und T_{gelb} erstellt, die jeweils Ähren eines frühen (Ähren mit grüner Färbung) und eines späten Reifestadiums (Ähren mit gelber Färbung) beinhalten. Jeder Datensatz besteht aus 250 Bildern mit einer Größe von 1531 x 1149 Pixeln. Durch die Vereinigung beider Datensätze wird ein weiterer Datensatz $T_{grün} \cup T_{gelb}$ erstellt, der Bilder beider Reifegrade beinhaltet.

Übertragbarkeit der synthethischen Daten auf reale Daten

Tabelle 1: Gütemaße für die Ährenerkennung basierend auf verschiedenen syntethischen Datensätzen.

Datensatz	IoU	Gesamt-Genauigkeit [%]	Präzision [%]	Sensitivität [%]
U-Net				
$T_{grün}$	46.21	87.71	64.75	61.73
T_{gelb}	43.67	86.65	63.08	58.66
$T_{grün} \cup T_{gelb}$	47.03	88.21	63.78	64.16
DeepLab				
$T_{grün}$	63.52	92.23	82.40	73.49
T_{gelb}	52.00	91.27	84.16	57.63
$T_{grün} \cup T_{gelb}$	69.96	93.88	86.84	78.25

Basierend auf den erzeugten Trainingsdaten werden die Gewichte der Netze gelernt. Aufgrund der begrenzten Speicherkapazität der GPUs werden die Bilder in insgesamt 7500 Patches einer Größe von 256 x 256 Pixeln unterteilt. Um eine möglichst große Robustheit der Netze zu erreichen, werden die Trainingsdaten generalisiert, indem die einzelnen Patches zufällig rotiert und vertikal oder horizontal gespiegelt werden. Die Datensätze werden jeweils zu 70 % als Trainingsdaten und zu 30 % als Testdaten verwendet. Für die verwendeten Netze werden bei den jeweiligen Testdatensätzen Gesamtgenauigkeiten von über 95 % erzielt.

Um die Übertragbarkeit der synthetischen Trainingsdaten zu analysieren, werden die trainierten Netze auf einen Datensatz bestehend aus 20 realen Bildern angewendet. Die Weizenpflanzen in den Auf-

nahmen weisen dabei unterschiedliche Reifegrade auf. Zu jedem Bild ist eine manuell erstellte Referenzmaske vorhanden.

Die Ergebnisse der Analyse sind in Tabelle 1 zusammengefasst. Als Maß für die Ähnlichkeit zwischen den prädizierten Masken und den Referenzmasken dient der Jaccard-Koeffizent (auch als Intersection over Union (IoU) bezeichnet). Zusätzlich ist die Gesamtgenauigkeit der Segmentierung, sowie die Präzision und die Sensitivität angegeben. Letztere beschreibt den Anteil der korrekt als Ähre erkannten Pixel gegenüber aller prädizierten Ährenpixel. Die Präzision liefert eine Aussage darüber, wie viele der in den Referenzmasken enthaltenden Pixel tatsächlich detektiert wurden.

Beim Vergleich der Ergebnisse der Datensätze fällt auf, dass $T_{\text{grün}}$ und T_{gelb} deutlich niedrigere Werte erzielen als bei ihrer Vereinigung $T_{\text{grün}} \cup T_{\text{gelb}}$. Es zeigt sich, dass die Modellierung verschiedener Reifestadien zu einer besseren Erkennung der Ähren führt. Des Weiteren fällt auf, dass die Werte der Präzision für $T_{\text{grün}}$ und $_{\text{gelb}}$ zwar ungefähr gleich sind, die Sensitivität bei T_{gelb} aber deutlich geringer ausfällt. Diese Unterschiede können dadurch erklärt werden, dass die verschiedenen Reifegrade bei den realen Bildern nicht gleichmäßig verteilt sind, sondern überwiegend Aufnahmen von grün gefärbten Pflanzen untersucht wurden. Die verschiedenen Netze beeinflussen das Ergebnis maßgeblich. Während die Ergebnisse des DeepLabs eine gute Erkennungbarkeit der Ähren belegen (der maximale IoU beträgt 69.96), weist das U-Net mit einem maximalen IoU von 47.03 eine deutlich geringe Erkennungsrate auf. Die beschriebenen Effekte lassen sich auch visuell in Abbildung 4.2 für die Auswertung von $T_{\text{grün}} \cup T_{\text{gelb}}$ erkennen. Zusätzlich ist für jedes Bild der jeweilige IoU angegeben. An diesem lässt sich erkennen, dass nicht alle Reifegrade mit derselben Güte erkannt werden. Die untersuchten realen Bilder weisen unterschiedlichste Reifestadien auf, wohingegen die synthetischen Datensätze sich nur auf zwei manuell modellierte Reifegrade stützen. Diese Diskrepanz ist vermutlich die Ursache für die variierende Erkennungsrate in den realen Aufnahmen.

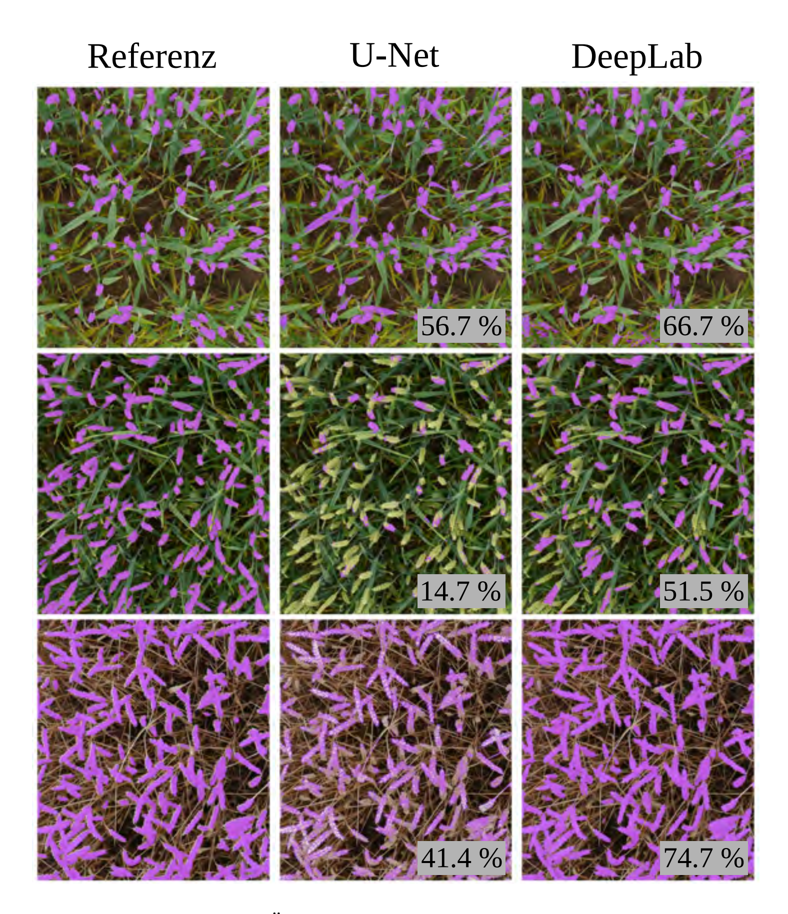


Abbildung 4.2: Ergebnis der Ährenerkennung für verschiedene reale Bildaufnahmen. Die erkannten Ähren sind violett markiert. Zusätzlich ist der IoU-Koeffizent jedes Bildes angegeben.

5 Fazit und Ausblick

Das Ziel der Arbeit war es, Weizenähren innerhalb von Farbbildern mittels neuronaler Netze zu erkennen. Anstelle manuell annotierter Trainingsdaten wurde dabei auf synthetisch erzeugte Daten zurückgegriffen. Die Ergebnisse zeigen, dass die Information aus

synthetisierten Daten auf reale Daten transferiert werden kann. Bestehende Abweichungen sind vor allem auf die nur geringe Anzahl an manuell modellierten Reifegraden innerhalb der Trainingsdaten zurückzuführen. In zukünftigen Arbeiten sollte daher eine automatisierte Modellierung verschiedener Wachstumsphasen angestrebt werden, um so ein größeres Spektrum an Informationen innerhalb der Trainingsdaten zu generieren.

Literatur

1. Y. Zhu, Z. Cao, H. Lu, Y. Li, and Y. Xiao, “In-field automatic observation of wheat heading stage using computer vision,” *Biosystems Engineering*, vol. 143, pp. 28 – 41, 2016.
2. E. David, S. Madec, P. Sadeghi-Tehran, H. Aasen, B. Zheng, S. Liu, N. Kirchgessner, G. Ishikawa, K. Nagasawa, M. A. Badhon, C. Pozniak, B. de Solan, A. Hund, S. C. Chapman, F. Baret, I. Stavness, and W. Guo, “Global wheat head detection (gwhd) dataset: a large and diverse dataset of high resolution rgb labelled images to develop and benchmark wheat head detection methods,” 2020.
3. C. Zhou, D. Liang, X. Yang, H. Yang, J. Yue, and G. Yang, “Wheat ears counting in field conditions based on multi-feature optimization and twsvm,” *Frontiers in Plant Science*, vol. 9, p. 1024, 2018.
4. T. Alkhudaydi, D. Reynolds, S. Griffiths, J. Zhou, and B. Iglesia, “An exploration of deep-learning based phenotypic analysis to detect spike regions in field conditions for uk bread wheat,” *Plant Phenomics*, vol. 2019, pp. 1–17, 07 2019.
5. J. Ma, Y. Li, K. Du, F. Zheng, L. Zhang, Z. Gong, and W. Jiao, “Segmenting ears of winter wheat at flowering stage using digital images and deep learning,” *Computers and Electronics in Agriculture*, vol. 168, p. 105159, 2020.
6. P. Sadeghi-Tehran, N. Virlet, E. Ampe, P. Reyns, and M. Hawkesford, “Deepcount: In-field automatic quantification of wheat spikes using simple linear iterative clustering and deep convolutional neural networks,” *Frontiers in Plant Science*, vol. 10, p. 1176, 09 2019.
7. Z. Grbović, M. Panić, O. Marko, S. Brdar, and V. Crnojevic, “Wheat ear detection in rgb and thermal images using deep neural networks,” 10 2019.

8. S. Ghosal, B. Zheng, S. Chapman, A. Potgieter, D. Jordan, X. Wang, A. Singh, A. Singh, M. Hirafuji, S. Ninomiya, B. Ganapathysubramanian, S. Sarkar, and W. Guo, “A weakly supervised deep learning framework for sorghum head detection and counting,” vol. 2019, 06 2019.
9. A. Tsirikoglou, J. Kronander, M. Wrenninge, and J. Unger, “Procedural modeling and physically based rendering for synthetic data generation in automotive applications,” *CoRR*, 2017.
10. S. Khan, B. Phan, R. Salay, and K. Czarnecki, “Procsy: Procedural synthetic dataset generation towards influence factor studies of semantic segmentation networks,” in *CVPR Workshops*, 2019.
11. S. I. Nikolenko, “Synthetic data for deep learning,” 2019.
12. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *CoRR*, vol. abs/1505.04597, 2015.
13. L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder ith atrous separable convolution for semantic image segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818.

Robuste kameragestützte Präzisionslandung von automatisierten fliegenden Systemen

Endres Kathe, Amilcar do Carmo Lucas, Enrico Neumann und Pedro Miguel Isaac Delso

IAV GmbH
Carnotstraße 1, 10587 Berlin

Zusammenfassung Eine besonders kritische Flugphase bei der Automatisierung von unbemannten automatisch fliegenden Systemen stellt das Landen dar. Je nach Größe des nutzbaren Flugkorridors und des Landeplatzes können hier Genauigkeiten im Zentimeterbereich an die Wiederholbarkeit der Flugbahn und Landeposition gefordert werden. Auch elektromagnetische Störungen können die Nutzung herkömmlicher Systeme wie GNSS besonders im Landebereich erschweren. Das von der IAV entwickelte optische Landesystem stellt eine ganzheitliche Entwicklung dar, die über die Landeplattform, die Einbindung und hardwareseitige Steuerung der Kamera, die Bildverarbeitung und Positionsberechnung bis hin zu der Integration in das Flugsteuerungssystem reicht. Durch den spezifischen Aufbau unseres Systems wird eine hohe Robustheit gegenüber Änderungen der Umgebungsbedingungen erreicht. Zusätzlich wird die Integrität des Systems durch die Schätzung der (Positions-)Genauigkeiten und der Rückmeldung des Zustandes des Gesamtsystems sichergestellt. Die Entwicklung und Tests des Systems erfolgten sowohl in der Simulation als auch unter verschiedenen realen Flugbedingungen.

Keywords Drohne, UAV, Automatisierung, Präzises Landen, Fiducial Markers, Robotik, Computer Vision, Robuste Schätzverfahren

1 Einführung

Drohnensysteme finden vermehrt Einzug in den kommerziellen Sektor. Damit verbunden steigt auch der Wunsch nach einem höheren Automatisierungsgrad. Eine besonders kritische Flugphase bei der Automatisierung stellt dabei das Landen dar. Je nach Größe des nutzbaren Flugkorridors und des Landeplatzes können hier Genauigkeiten im Zentimeterbereich an die Wiederholbarkeit der Flugbahn und Landeposition gefordert werden. Zusätzlich kann es gehäuft im Landebereich zu Abschattungen [1] und/oder Multipath [2] von GNSS-Signalen, wie etwa durch hohe Wände, kommen. Aber auch andere Störgrößen sind oftmals auf den letzten Metern des Fluges anzutreffen, wie z. B. magnetische oder elektrische Einflüsse durch Stromleitungen. Dies behindert oft das Nutzen herkömmlicher Positions- und Orientierungssysteme wie z. B. GPS-Empfänger oder Magnetometer.

Kameras sind von diesen Störungen nicht betroffen. Zusätzlich stellen sie eine informationsreiche Sensorquelle dar, die hohe Redundanzen und Genauigkeiten ermöglicht. Deshalb haben sich die optischen Verfahren für Landeanflüge als besonders geeignet herausgestellt.

Im Folgenden soll das durch die IAV GmbH entwickelte System vorgestellt werden, was den Fokus auf Robustheit mittels Redundanz, Integrität und Flexibilität setzt, welche nachstehend erläutert werden.

2 Beschreibung des Landesystems

Das System besteht aus einer aktiven Komponente, die sich auf der Drohne befindet. Diese besteht aus einer Industriekamera mit global Shutter im Verbund mit einem Companion-Computer, auf welchem die Software läuft. Zusätzlich kann an der Drohne eine Beleuchtung angebracht werden, welche eine Landung bei Nacht ermöglicht. Ein Teil des Systems ist in Abbildung 2.1 dargestellt.

Die passive Komponente befindet sich am Landeplatz und besteht aus einer Folienkombination aus retroreflektivem weiß und mattem schwarz. Durch das matte schwarz werden Reflektionen durch



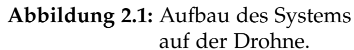


Abbildung 2.1: Aufbau des Systems auf der Drohne.

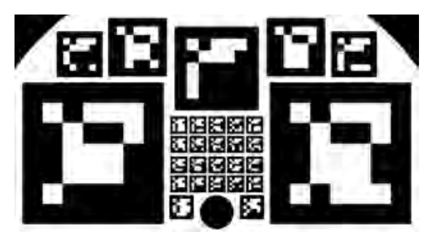


Abbildung 2.2: Aufbau der Landemarkierung.

ungünstige Einstrahlwinkel der Sonne oder anderer Lichtquellen minimiert. Das retroreflektive Weiß ermöglicht das effiziente Beleuchten durch die Drohne bei Dunkelheit. Die Folien bilden Aruco-Marker [3] nach dem Muster, wie es in Abbildung 2.2 ersichtlich ist. Diese werden in ihrer Größe und Kombination der Landefläche angepasst, um die Sichtbarkeit mehrerer Marker bei unterschiedlichen Höhen im Kamerabild zu ermöglichen.

2.1 Beschreibung der Bildverarbeitung

Das in diesem Artikel vorgestellte Verfahren kombiniert verschiedene Algorithmen der Bildverarbeitung zu einer Datenverarbeitungskette, um aus den aufgenommenen Bildern zunächst die Markierungen zu erkennen und anschlißened eine Lösung der relativen Position und Orientierung zu erhalten. Zusätzlich wird die Güte der Positionslösung geschätzt.

Da das System im Außenbereich arbeitet, ist es starken Schwankungen des Umgebungslichts ausgesetzt, welche von einer dunklen Nacht bis zu einem hellen, wolkenfreien Sommertag mit gegebenfalls ungleichmäßiger Beleuchtung reichen.

Einstellung des Kamera-Gains

Das Hardware-Gain der Kamera wird je nach Umgebungslicht mittels einer Histogrammanalyse dynamisch angepasst. Die Verschlusszeit wird auf einen empirisch ermittelten Wert fixiert und stellt einen Kompromiss zwischen Bildrauschen und Motion-Blur dar, der durch die Flugbewegung verursacht wird.

Adaptive Threshold

Das durch die Kamera erzeugte Bild wird zuerst mit einem Adaptive Threshold Algorithmus binarisiert, um es somit in das Schwarzweiße zu übersetzen. Hier wird bei jedem Pixel ein Histogramm der Nachbarschaftspixel erstellt, wodurch der individuelle Schwellwert für die Binarisierung berechnet wird [4]. Wird bei den nachfolgend erläuterten Prozessen kein Marker im Bild erkannt, wird der Parameter der Nachbarschaftsgröße stetig automatisch durch das System verändert um so unterschiedlichste Ausleuchtungen der Landemarkierung kompensieren zu können.

Konturensuche

Das binarisierte Bild wird nun nach konvexen Vierecken durchsucht. Zuerst werden Konturen mit einem border following algorithmus gefunden [5]. Anschließend werden diese mit dem Verfahren von Douglas-Peucker vereinfacht [6].

Danach kommt das Quadrilateral Sum Conjecture-Kriterium zum Einsatz um zu prüfen, ob es sich bei einer gefundenen Kontur um ein konvexes Viereck bzw. ein Quadrat unter perspektivischer Verformung handelt [7]. Hierbei muss die Summe der vier Winkel der Kontur 360° ergeben. Dieses Kriterium ermittelt auch Quadrate unter starker perspektivischer Verzerrung. Als weitere Kriterien darf die Summe des Cosinus der Winkel des Vierecks einen gewissen Wert nicht überschreiten und die Pixelfläche des gefundenen und auf die Bildfläche projektierten Viereck muss eine Mindestgröße besitzen, um so Rauschen herauszufiltern.

Identifikation der Aruco-Marker

Wurde ein Viereck im Bild gefunden, folgt die Prüfung, ob es sich um einen Aruco-Marker [3] handelt. Zur Verwendung kommt hierbei das standard Aruco-Dictionary [8]. Die perspektivische Verzerrung wird korrigiert. Anschließend wird mit Hilfe einer linearen Interpolation das Bild in eine 7x7 Matrix übersetzt, welche anschließend mit dem Otsu-Binarisierungsalgorithmus [9] wieder in das binäre übersetzt wird. Die gefundene Binärmatrix wird auf die Zugehörigkeit des Aruco-Codebreichs mit Hilfe der Signaturmatrix geprüft. Anschließend werden die auf das Bild projektierten gefundenen Eckpunkte des Markers Koordinaten im Raum zugeordnet, welche zuvor eingemessen wurden und die in einer Datenbank hinterlegt sind.

Gewinnung der Positionslösung

Anschließend wird versucht, mit Hilfe des PnP-Algorithmus von OpenCV, welcher auf der iterativen Levenberg-Marquardt Optimierung basiert, die Pose der Kamera relativ zum Marker zu finden.

Daraufhin wird der Reprojektionsfehler berechnet. Überschreitet dieser einen gewissen Schwellenwert, so wird davon ausgegangen, dass das iterative Lösungsverfahren fehlgeschlagen ist. Dies kann zum Beispiel der Fall bei einer Fehldeketion sein, ein Marker wurde fälschlich identifiziert oder aber es wurde der Versuch von spoofing unternommen, indem ein ein weiterer Marker aus dem gleichen Code-Bereich in das Bild gebracht wurde. Die Lösung des PNP-Problems wird dann noch einmal mit dem RANSAC-Verfahren unternommen. Dieses Verfahren weist eine höhere Robustheit gegen Außreißer auf, indem es diese aus der Lösung ausschließt.

Anschließend erfolgt eine Schätzung der Güte C der Positionslösung. Nach dem hier vorgestellten Modell verhält sich diese antiproportional zu der Fläche A eines konvexen Polygons, welche die Projektion der Marker auf die Bildfläche einhüllt und proportional zu dem Abstand d zwischen Kamera und Markermitte. Ein zusätzlicher, linearer Faktor k wird empirisch durch die Simulation ermittelt und spiegelt die intrinsischen Parameter der Kamera wieder.

$$C = k \cdot \frac{d}{A} \tag{2.1}$$

Durch den redundanten Aufbau der Landemarkierung kann trotz des Ausfalls eines oder mehrerer Marker eine Positionslösung generiert werden und eine Landung erfolgen. Darüber hinaus gibt es eine Rückmeldung darüber, welche Marker bei einem Landeanflug nicht erkannt wurden, wie es zum Beispiel bei einer Verdeckung der Fall ist. So kann der Zustand der Markierung von dem System selbst verfolgt und bei Bedarf von Mensch eingeschritten werden um zum Beispiel die Markierung zu reinigen oder zu erneuern.

Integration der Positionslösung in den Flugcontroller

Als letzter Schritt folgt die Integration der Positionslösung in dem Flugcontroller, um die präzise Landung zu ermöglichen. Für die Software des Flugcontrollers erfolgt die Wahl des Flightstacks Arducopter.

Die Standardimplementierung für die Präzisionslandung erwartet hierbei als Eingang einen Winkel der Line of Sight zwischen der optischen Achse der Kamera und der Landemarkierung, sowie deren Distanz zueinander. Auf der Drohne werden ständig die Lagewinkel der Drohne mit Hilfe der Intertial Measurement Unit (IMU) gemessen. Da die Kamera fest mit der Drohne verbunden ist, ist somit auch die Orientierung der Kamera bekannt. Die gemessenen Winkel der Kamera können zu einem Einheitsvektor übersetzt werden, welcher von dem Körperfesten Koordinatensystem der Kamera zu der Landemarkierung zeigt.

$$||\vec{v}_{body_unit}|| = 1 \tag{2.2}$$

Mit Hilfe einer Transformationsmatrix kann der Vektor von dem körperfesten Koodrinatensystem in das North-East-Down (NED)-Koordinatensystem überführt werden.

$$\vec{v}_{ned_unit} = T_{ned_body} \cdot \vec{v}_{body_unit} \tag{2.3}$$

Anschließend wird der Einheitsvektor mit der Distanz zu der Landeposition multipliziert, um so einen Vektor zu erhalten, der die Ablage der Drohne von der Landeposition im NED-Koordinatensystem beschreibt.

$$\vec{v}_{ned} = \vec{v}_{ned_unit} \cdot d_{target_distance} \tag{2.4}$$

Dieses Verfahren ist besonders in größeren Höhen geeignet, wo die Güte der Positionslösung nach Gleichung 2.1 durch die geringe Größe der Landemarkierung im Bild und den hohen Abstand beeinträchtigt ist.

In niedrigeren Höhen steigt die Güte der Positionslösung. Unterschreitet diese einen Schwellenwert, so kann die Ablageposition von Drohne zu Landemarkierung direkt und ohne Umwege aus der Markierung mit Hilfe der Lösung des PnP-Problems gelesen und in den Flugcontroller eingespeist werden. Hierür wurde der ArduCopter-Code modifiziert und es entfallen Fehler, die durch Messfehler der IMU, Ungenauigkeiten bei der IMU-Kamera synchronisierung oder bei einem schräglagigen Einbau der Kamera entstehen würden.

3 Simulation

Für die Simulation wurde die Umgebung Gazebo [10] genutzt. Dabei wurden die intrinsischen Parameter der Kamera mit einem horizontalen FOV von 68° bei einer Auflösung von 1216x1024 Pixel und der Aufbau der optischen Markierung nach Abbildung 2.2 mit einer Breite von 1,4m nachmodelliert. Die Pose zwischen Marker und Kamera wurde nach dem Zufallsverfahren variiert. Anschließend berechnet der Algorithmus die Position zwischen Kamera und Landemarkierung sowie den Winkel der Line Of Sight (LOS) der Kamera. Bei jeder Iteration wird die Lösung aus der Bildverarbeitung zusammen mit der Groundtruth aus der Simulation gespeichert. Hierdurch ist es möglich, den Messfehler zu bestimmen, welcher in der Tabelle 1 und 2 sowohl für die Position als für die Winkelbestimmung dargestellt ist. Die Positionslösung ist hierbei bereits nach dem Gütekriterium gefiltert und daher nur in einer Höhe bis zu 2m vorhanden.

Tabelle 1: Positionsgenauigkeit in verschiedenen Höhenbändern

Höhenband [m]	Anzahl der Messpunkte	$\mu \pm 1\sigma[m]$
0 - 1	42	0.0022 ± 0.0065
1 - 2	108	0.0021 ± 0.0207

Tabelle 2: Winkelgenauigkeit in verschiedenen Höhenbändern

Höhenband [m]	Anzahl der Messpunkte	$\mu \pm 1\sigma[rad]$
0 - 1	45	0.0109 ± 0.0180
1 - 2	124	0.0024 ± 0.0058
2 - 3	141	0.0017 ± 0.0027
3 - 4	135	0.0012 ± 0.0023
4 - 5	150	0.0013 ± 0.0010

Das Gütekriterium wurde empirisch ermittelt und eliminiert Außreißer in der Positionslösung. In einer Höhe von unter 2m werden 88,76% der Positionslösungen direkt in den Flugcontroller eingespeist.

Die Winkelmessung ist über alle gemessene Höhenbänder nach Tabelle 2 stabil.

4 Reale Testflüge

Bei dem Testsystem handelt es sich um eine gefesselte Drohne, bei der die Stromversorgung aber auch die Datenübertragung über ein zu einem Hangar geführtes Kabel dargestellt wird. Das Kabel wird über ein mechatronisches System nachgeführt und gestrafft, damit dieses im Flug nicht durchhängt.

Das System fungiert als fliegende Überwachungskamera. Bei einem Testablauf wird ein Einbruch simuliert. Der Hangar öffnet sich, die gefesselte Drohne hebt ab und fliegt zu der Position des Einbruchs, um diesen zu filmen. Anschließend schwebt die Drohne zurück über den Hangar und beginnt den Landeanflug. In einer Höhe von ca. 13m ist die Landemarkierung in dem Kamerabild ersichtlich und es beginnt das präzise Landen auf dem Hangar.

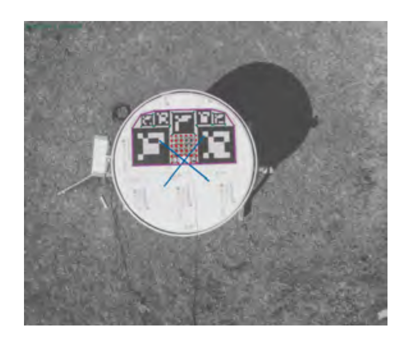




Abbildung 4.1: Sicht der Landekamera im Flug



Abbildung 4.2: Kugelhangar beim Schließvorgang

Durch die Dimensionierung der Drohne und des Hangars darf die Abweichung von der anvisierten Landeposition eine absolute Abweichung von ca. 20cm nicht überschriten, da dieser ansonsten nicht schließen kann.

Im Folgenden sollen die Ergebnisse von 11 Testflügen untersucht werden, die mit dem System unternommen wurden.

Es zeigt sich eine Standardabweichung von 0,045m bei einem Erwartungswert von 0,039m Abweichung. Die maximale Abweichung der Landeposition beträgt 0.104m. Es wurde mit Windgeschwindigkeiten von bis zu 7m/s geflogen.

5 Zusammenfassung

In diesem Beitrag wurde ein optisches Verfahren für die Präzisionlandung von automatischen fliegenden Systemen vorgestellt.

Dieses Verfahren ermöglicht eine hohe Verfügbarkeit, indem es sich durch das automatische Einstellen der Software- und Hardwareparameter an die Belichtungszustände, die bei einem Einsatz im Außenbereich vorzufinden sind, anpasst.

Das System ist derart redundant aufgebaut, so dass es sehr robust gegen partiellen Verschleiß oder Verdeckung von Markern beispielsweise infolge der Witterung reagiert. Ein Ausfall von einzelnen Markern kann vom System erkannt und gemeldet werden.

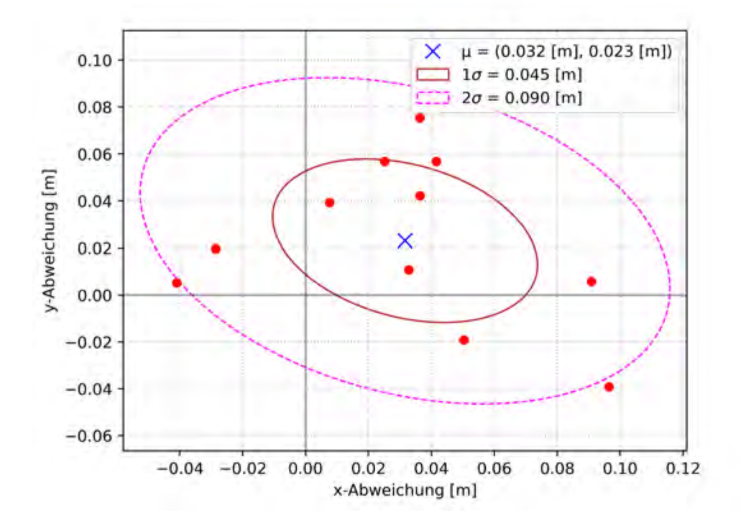


Abbildung 4.3: Positionsabweichung der Drohne von der Ziellandeposition

Durch die Prüfung der Kriterien für die Positionsgüte kann das System eine Rückmeldung über die eigene Integrität geben. Anhand der Positionsgüte wird zwischen zwei Verfahren gewählt, um die Landeplatz-Position in das Flugsteuerungssystem einzuspeisen. Das erste Verfahren kommt bei schlechter Sichtbarkeit der Landemarkierung zum Einsatz, wie es in größeren Höhen der Fall sein kann und integriert Sensormessungen des Flugsteuerungssystem in die Positionslösung, um eine robuste Positionslösung zu erhalten. Das zweite Verfahren kommt bei einer guten Sichtbarkeit der Landemarkierung zum Einsatz, wie es auf den letzten Metern des Landeanfluges der Fall ist. Hier kann die Positionsabweichung direkt aus der Markierung abgeleitet werden und der Fokus liegt auf einer hohen Präzision um ein zentimetergenaues Landen zu ermöglichen.

Simulationsergebnisse zeigen bei der Positionslösung auf dem für die Landepräzision besonders relevanten Höhenband von 0m-1m einen Erwartungswert von 0.0022m mit einer Standardabweichung von 0.0065m.

Elf reale Flugversuche unter Windeinfluss mit bis zu 7m/s zeigten hierbei eine maximale Abweichung der Landeposition von 0.104m. Der Erwartungswert liegt bei 0,039m bei einer Standardabweichung von 0,045m.

Das System eignet sich somit zum automatischen Landen in einem Dronenhangar, der ganzjährig und zu jeder Tageszeit betrieben wird und findet somit Einsatz bei verschiedenen Projekten der IAV GmbH.

Literatur

1. F. Zimmermann, C. Eling, L. Klingbeil, and H. Kuhlmann, “Precise Positioning of Uavs - Dealing with Challenging Rtk-Gps Measurement Conditions during Automated Uav Flights,” *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42W3, pp. 95–102, Aug. 2017.
2. T. Kos, I. Markezic, and J. Pokrajcic, “Effects of multipath reception on gps positioning performance,” in *Proceedings ELMAR-2010*, 2010, pp. 399–402.
3. S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, vol. 47, p. 2280–2292, 06 2014.
4. M. Sezgin *et al.*, “Survey over image thresholding techniques and quantitative performance evaluation,” *Journal of Electronic imaging*, vol. 13, no. 1, pp. 146–168, 2004.
5. S. Suzuki and K. be, “Topological structural analysis of digitized binary images by border following,” *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32 – 46, 1985. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0734189X85900167
6. S.-T. Wu, A. C. G. d. Silva, and M. R. G. Márquez, “The Douglas-peucker algorithm: sufficiency conditions for non-self-intersections,” *Journal of the Brazilian Computer Society*, vol. 9, pp. 67 – 84, 04 2004. [Online]. Available: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-65002004000100006&nrm=iso
7. J. Ferrão, P. Dias, and A. Neves, “Detection of aruco markers using the quadrilateral sum conjuncture,” *Lecture Notes in Computer Science*, vol. 10882, pp. 363–369, 06 2018.

8. S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognition*, vol. 51, pp. 481 – 491, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320315003544
9. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
10. N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep 2004, pp. 2149–2154.

Bildbasierte Geolokalisierung für UAVs

Michael Schleiss

Fraunhofer FKIE,
Fraunhoferstr. 20, 53343 Wachtberg

Zusammenfassung When unmanned aerial vehicles (UAVs) fly autonomous missions, they typically rely on global satellite navigation systems (GNSS) like GPS for global position estimation. However, GNSS signals can be easily jammed. We propose a camera-based method that uses onboard imagery and data from OpenStreetMap as a backup system for GNSS. First, the aerial imagery from the onboard camera is translated into a map-like representation. Then we match it with a reference map to infer the vehicle's position. Experiments over a typically sized mission area are performed and exhibit localization accuracy close to 6 m. Our results show that the proposed method can serve as a backup to GNSS systems where suitable landmarks like buildings and roads are available.

Keywords Image-based navigation, geolocalisation, GPS-denied, UAV

1 Einleitung

Wenn selbstfahrende Autos durch Tunnel oder tiefe Hochhausschluchten fahren, dann benötigen diese beim Navigieren einen Ersatz für die Satellitennavigation, denn GPS und Co stehen in diesen Situationen nicht zur Verfügung. Ähnlich sieht es beim Einsatz von autonomen UAVs in geschlossenen Räumen aus. Daher wurden für diese Einsatzzwecke unter anderem visuelle Methoden zur Lokalisierung erforscht [1, 2], die bei fehlendem Signal von Satellitennavigationssystemen (GNSS) als Ersatz fungieren können.

Anders sieht es beim Außeneinsatz von autonomen Drohnen aus. Hoch in der Luft geht man bisher von einem sehr guten Empfang von GNSS-Signalen und einer hohen Genauigkeit der Eigenpositionsbestimmung (1-5 m) aus [3]. Mit Hilfe im Handel frei verfügbarer Technik kann man jedoch GNSS-Signale blockieren (jamming) oder fälschen (spoofing) [4]. UAVs, die zur Lokalisierung nur auf GNSS setzen, werden so zum Landen gezwungen und können von böswilligen Akteuren gekapert oder zerstört werden. Im schlimmsten Fall droht sogar der Absturz des Vehikels.

Wenn man jedoch bedenkt, dass autonome Luftfahrzeuge in Zukunft ein integraler Bestandteil der Logistik werden und den Transport von wertvollen Güter, wie Medikamente [5] und Organe [6], oder sogar Personen [7] automatisieren sollen, wird schnell klar, dass man auch beim Einsatz von UAVs unter freiem Himmel eine Backup-Strategie für den Ausfall der Satellitennavigation benötigt. Auch Polizei- und Rettungskräfte werden in der Zukunft vermehrt auf den Einsatz von UAVs zurückgreifen, zum Beispiel zur Bewachung von kritischer Infrastruktur, der Verschaffung eines überblicks bei Katastrophen oder der Suche nach Vermissten. Auch hier liegt eine Verletzbarkeit vor, die durch Kriminelle und Terroristen ausgenutzt werden könnte.

Ziel der Forschungstätigkeit ist es daher eine Methode zur visuellen Bestimmung der Eigenposition für UAVs unter freiem Himmel vorzustellen, um GNSS-Jamming und Spoofing umgehen zu können (siehe Abb. 1.1).

2 Verwandte Arbeiten

Wir unterscheiden zunächst einmal zwei Arten der Lokalisierung. Die relative Lokalisierung gibt die Position in Bezug auf einen Startpunkt an, bei der absoluten Lokalisierung erhält man eine georeferenzierte Position in Latitude und Longitude, so wie bei einem Satellitennavigationsempfänger.

Als Beispiel für die relative Lokalisierung sei die visuelle Odometrie genannt mit Hilfe des optischen Flusses genannt [8]. Es werden aufeinanderfolgende Bildpaare verglichen und, vorausgesetzt man kennt die Flughöhe, aus dem Versatz eine Bewegungsrichtung und



Abbildung 1.1: Aufgrund der hohen Distanz zwischen UAV und den Navigationssatelliten ist es ein Leichtes, das Signal mit Hilfe eines Störers zu stören. Die Nutzung der vorhandenen Bordsensorik soll in diesem Fall eine autarke Lokalisierung ermöglichen.

Geschwindigkeit berechnet. Bei der absoluten Lokalisierung kommen zum Beispiel im militärischen Umfeld traditionell terrainbasierte Lokalisierungsverfahren oder Scene Matching-Verfahren, wie TERCOM [9] und DSMAC [10], zum Einsatz.

Die weite Verbreitung von kommerziell erhältlichen UAVs in der jüngeren Vergangenheit führte auch zu vermehrter Forschung zu visueller Geolokalisierung in diesem Bereich. Einer der ersten Versuche eine Geolokalisierung aus den Bildern der Bordkamera eines leichtgewichtigen UAVs zu gewinnen, wurde von Conte und Doherty vorgeschlagen [8]. Diese kombinieren eine visuelle Odometrie mit einem Algorithmus, der die Bordbilder mit einer Datenbank aus georeferenzierten Luftaufnahmen abgleicht, um den Drift zu reduzieren. Der Abgleich basiert auf der normalisierten Kreuzkorrelation der Bildintensitäten. Conte und Doherty berichten von brauchbaren Ergebnissen, dies ist jedoch überwiegend auf die Leistung der visuellen Odometrie zurückzuführen. Das Modul für die Driftkorrektur liefert nur verhältnismäßig seltene Ausgaben, da die meisten „Matches“ wegen hoher Unsicherheit zurückgewiesen werden. In ih-

rem Experiment konnten nur an zwei Positionen eine Driftkorrektur durchgeführt werden [8].

Im Gegensatz dazu nutzen Cesetti et al. Feature Deskriptoren, nämlich SIFT, für die Georeferenzierung der Bordbilder [11]. Vorausgesetzt werden große Flughöhen, um sinnvolle Merkmale aus natürlichen Landmarken extrahieren zu können. In geringen Flughöhen können verrauschte Muster von Bäumen, Wiesen etc. im Bildmaterial dominieren. In ihren Experimenten berücksichtigen Cesetti et al. daher nur Bordbilder mit einem Fußabdruck am Boden von mindestens einem Quadratkilometer. Dadurch ist der Einsatz dieser Methode nur auf spezifische Szenarien eingeschränkt.

Grönwall et al. erweitern [8] indem Sie Lidar-gestützte Messungen für die visuelle Odometrie heranziehen [12]. Jedoch bleibt das grundlegende Problem von seltenen Matches weiterhin bestehen. Shan et al. übersetzen Bilder und Referenzmaterial in eine HOG-basierte (Histogram of oriented Gradients) Repräsentation.

Lindsten et al. segmentieren das Bild anhand verschiedener Klassen wie Straßen, Gebäude, Wiesen und Gewässern und vergleichen dann mit Hilfe eines Histograms der Pixelhäufigkeiten pro Klasse die Bildaufnahmen mit einer entsprechenden Referenzkarte [13]. Zur Segmentierung kommt ein Clusteringalgorithmus in Form von Superpixeln zum Einsatz. Durch die Nutzung eines Histograms gehen jedoch geometrische Informationen verloren und führen zu uneindeutigen Positionsschätzungen in Arealen mit ähnlichen Klassenverteilungen.

Mannberg und Savvaris nutzen Objekt Detektoren, um die Position von Gebäuden in Luftaufnahmen zu bestimmen und reduzieren die Detektionen in eine Repräsentation, in der jedes Gebäude von einem Punkt auf einer Karte dargestellt wird [14]. Ein Fingerabdruck, der die geometrische Anordnung der Punkte berücksichtigt, wird berechnet und in einer Referenzdatenbank abgeglichen. Die Autoren berichten, dass ihr Framework auch auf andere Typen von Landmarken ausgeweitet werden kann. Es ist aber unklar, inwiefern Landmarken, die man nicht zu Punkten reduzieren kann, wie Straßen und Flüsse, eingebunden werden sollen.

Wir nutzen den gleichen Template Matching Ansatz wie Conte und Doherty. Aber wir erhalten höhere Matchingraten, indem wir die Bordbilder segmentieren und dadurch in eine robuste Re-

präsentation überführen. Der Segmentierungsprozess ist vergleichbar mit dem Ansatz von Lindsten, allerdings nutzen wir ein neuronales Netz und gleichen die segmentierten Bilder mit einer Referenzkarte statt einem Histogram. Dadurch bleibt die geometrische Anordnung der Landmarken erhalten. Unsere Methode funktioniert in verschieden Flughöhen typisch für kommerzielle UAVs und kann mehrere Arten von Landmarken (Häuser, Gebäude, Wälder, Flüsse, etc.) berücksichtigen.

3 Methodik

Ähnlich dem Scene Matching und dem Verfahren von Conte et al. haben wir ein Verfahren entwickelt, das das Bild einer zum Boden gerichteten Bordkamera mit einer Referenzdatenbank vergleicht und zur absoluten Lokalisierung nutzt, also georeferenzierte Positionen ausgibt. Das Verfahren lässt sich in drei Schritte einteilen (siehe auch Abb. 3.1).

- Die Bordkamera erstellt eine Luftaufnahme (Nadir).
- Ein neuronales Netz segmentiert die Luftaufnahme und übersetzt sie damit in eine straßenkartenähnliche Repräsentation.
- Das segmentierte Bild wird mit einer Referenzkarte bestehend aus Straßen und Hausgrundrissen per Template Matching abgeglichen.

Im Rahmen der visuellen Geolokalisierung ist es in unserem Fall ausreichend zwei Freiheitsgrade, nämlich Latitude und Longitude zu bestimmen, denn sowohl die Orientierung als auch die Höhe über Grund können driftfrei per inertialer Messeinheit, Magnetometer und Altimeter bestimmt werden. Entsprechende Sensorik ist für kommerziell erhältliche UAVs verfügbar und kann vorausgesetzt werden. Wir erwarten zudem, dass die Bordbilder nach Norden und in Lotrichtung zum Boden ausgerichtet sind. Dies kann durch ein 3-DoF Gimbal problemlos sichergestellt werden. Alternativ werden die Bilder mit Hilfe von Informationen der inertialen Messeinheit und eines magnetischen Kompasses perspektivisch korrigiert.

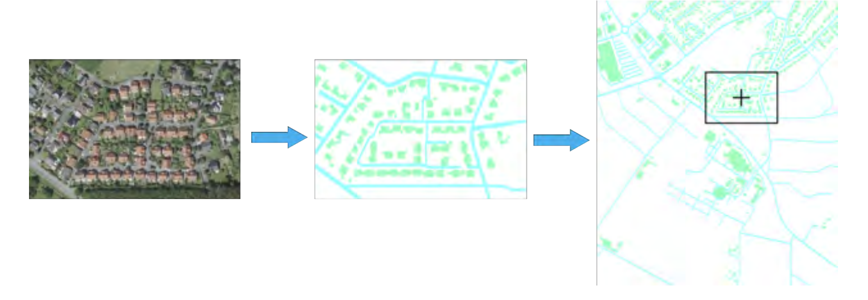


Abbildung 3.1: Das Verfahren besteht aus drei Schritten (v.l.n.r.): Aufnahme eines Luftbildes, Segmentierung von Straßen (blau) und Häusern (grün), Abgleich mit einer Referenzkarte.

Für die Segmentierung kommt ein neuronales Netz zum Einsatz, das darauf trainiert wurde, bestimmte Landmarken zu erkennen. Wir haben uns für Gebäude und Straßen entschieden, da diese bereits eine große Abdeckung in vielen Einsatzszenarien ermöglichen und entsprechendes Referenzmaterial öffentlich und kostenlos verfügbar ist, zum Beispiel über OpenStreetMap. Wir greifen dabei auf die U-Net Architektur zurück, die ursprünglich aus dem Bereich der medizinischen Bildverarbeitung stammt [15], mittlerweile aber in vielen anderen Szenarien, wie zum Beispiel der pixelweisen Segmentierung von Straßenszenen [16], zur Anwendung kommt. In unserem Fall empfängt das neuronale Netz die Bordbilder einer Tageslichtkamera und weist jedem Pixel eine Klasse, zum Beispiel Haus, Hintergrund oder Straße zu.

Wir nutzen, ähnlich wie Conte und Doherty [8] ein Template Matching Verfahren. Dabei wird das segmentierte Bild in Sliding-Window-Manier über die Referenzkarte geschoben. An jeder Position wird die Summe der Quadrate der Grauwertunterschiede bestimmt. Die Position mit der geringsten Abweichung stellt für uns einen Match dar und wird für die Ausgabe der Positionsschätzung herangezogen. Voraussetzung sind, dass der Maßstab, in dem die Bordbilder aufgenommen wurden, bekannt ist. Dieser wird mit Hilfe des Altimeters bestimmt und die Bordbilder dementsprechend skaliert, sodass sie mit der Bodenauflösung der Referenzkarte übereinstimmt.

Wir treffen in unserem Verfahren folgende Annahmen. Wir gehen davon aus, dass die Erdoberfläche lokal in unserem Missionsgebiet durch eine Ebene approximiert werden kann. Wir gehen auch davon aus, dass die grobe Position zu Beginn des Fluges bekannt ist, sodass das Flugvehikel mit einer geeigneten Referenzkarte des Missionsgebiets ausgestattet werden kann.

4 Training des Bildsegmentierers

Der Bildsegmentierer dient dazu, Landmarken wie Häuser und Gebäude aus den Bordbildern zu extrahieren. Dafür trainieren wir diesen mit Hilfe einer großen Sammlung an öffentlich verfügbaren Luftaufnahmen[1] und Daten aus OpenStreetMap. Wir haben dafür ein 125km^2 großes Gebiet um Bonn gewählt, das sowohl urbane auch ländliche Komponenten enthält. Das Gebiet wurde in Patches, der Größe 512 x 512 Pixel mit einer Bodenauflösung von 0,1 m pro Pixel aufgeteilt. Dies entspricht ca. 90.000 Trainingsbildern.

Die Trainingsmasken wurden mit Hilfe von Gebäudeumrissen und Straßenlinien aus OpenStreetMap erstellt. Da für Straßen nur die Mittellinie vermerkt ist, wurde die Breite anhand des Typs der Straße (Autobahn, Bundesstraße, Wohnstraße, etc.) geschätzt. Für die weiteren Details des Trainingsprozedere verweisen wir auf [17].

Die Luftaufnahmen bilden ein Gebiet über 100km^2 aus dem Stadtgebiet von Bonn ab. Es enthält den Stadtkern, aber auch Randgebiete, landwirtschaftliche Flächen und Wälder. Einmal trainiert lässt sich der Bildsegmentierer auch über anderen Gebieten anwenden, solange diese in ihrer Erscheinung dem Trainingsdatensatz ähneln [18,19].

5 Evaluierung

Das Lokalisierungsexperiment wird auf einem separaten Datensatz evaluiert. Anstatt der öffentlich verfügbaren Luftbilder, nutzen wir hier Daten, die wir selbst auf einem Gebiet südlich von Bonn aufgenommen haben.

[1] Digitale Orthophotos NRW: `https://www.bezreg-koeln.nrw.de/brk_internet/geobasis/luftbildinformationen/aktuell/digitale_orthophotos`

Die Nutzlast des Flugvehikels besteht aus einer Tageslichtkamera und einem INS. Die Kamera nimmt fünf Bilder pro Sekunde mit einer Auflösung von 3.280 × 2.464 Pixeln, einem Öffnungswinkel von 39,1° und einer durchschnittlichen Flughöhe von circa 300 m auf. Die Bodenfläche beträgt ca. 216 m auf 144 m bei einer Auflösung von unter 0,1 m pro Pixel.

Das inertialen Navigationssystem (INS) besteht aus Gyroskop, Accelerometer und Magnetometer. Ein Altimeter zur Höhenmessung steht in dieser Messreihe nicht zur Verfügung. Stattdessen wird die Höheninformation des GPS-Moduls genutzt. Die Kamera und das INS sind fest am Flügel fixiert und nicht an einem Gimbal angebracht. Neben Kamera und INS befindet sich ein GPS+RTK Modul in der Nutzlast, mit dessen Hilfe eine zentimetergenaue Position als Referenz für die Evaluierung aufgezeichnet wird.

Das Verfahren wird auf einer Flugbahn von 1,61km Länge evaluiert. Die Referenzkarte umfasst ein Gebiet von circa einem Quadratkilometer.



Abbildung 5.1: Links ist eine Luftaufnahme des Referenzgebiets zu sehen, in dem die Position gesucht wurde. Rechts sieht man die ermittelten Positionen (blau) und die Referenzpositionen, die durch das GPS+RTK ermittelt wurden (rot). Die drei blauen Punkte über dem Acker (links, Mitte) stellen Fehllokalisierungen dar.

Die Trajektorie besteht aus 471 Einzelbildern der Bordkamera. Jedes dieser Bilder wurde unabhängig in der Referenzkarte lokalisiert, um den Fokus auf die Qualität der Geolokalisierung zu legen. Das heißt, es wurde keine Informationen eines Bewegungsmodells berücksichtigt und keine Filterschritte durchgeführt.

Das Ergebnis der Evaluierung wird in Abb. 5.1 dargestellt. Die mittlere Abweichung von der Referenzposition beträgt 5,7m bei einer Standardabweichung von 7,4m.

6 Fazit

Bezogen auf die Positionierungsgenauigkeit ist das beschriebene Verfahren damit in der Lage, in diesem Anwendungsfall, als Ersatz für Satellitennavigationssystemen zu fungieren. Kritisch ist zu sehen, dass das Verfahren nur in Gebieten funktioniert, wo es auch ausreichend menschliche Bebauung in Form von Straßen und Häusern gibt. Ziel der weiteren Forschung wird es sein, diese Limitation aufzuheben. Zudem sollen weitere Datensätze gesammelt werden, die eine Vielzahl von Einsatzszenarien, wie zum Beispiel unterschiedliche Jahreszeiten, Wetterbedingungen oder Landbedeckungen abdecken und als Benchmark für die entwickelten Verfahren dienen sollen.

Literatur

1. J. V. Carroll, “Vulnerability assessment of the us transportation infrastructure that relies on the global positioning system,” *The Journal of Navigation*, vol. 56, no. 2, p. 185, 2003.
2. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
3. H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, “Inloc: Indoor visual localization with dense matching and view synthesis,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7199–7209.

4. A. Ruegamer, D. Kowalewski *et al.*, "Jamming and spoofing of gnss signals–an underestimated risk?!" *Proc. Wisdom Ages Challenges Modern World*, vol. 3, pp. 17–21, 2015.
5. B. McCall, "Sub-saharan africa leads the way in medical drones," *The Lancet*, vol. 393, pp. 17–18, 2019.
6. M. Francisco, "Organ delivery by 1,000 drones," *Nature Biotechnology*, vol. 34, p. 684, 2016.
7. D. Lenton, "The measure of volocopter flying taxi," *Engineering & Technology*, vol. 13, no. 7/8, pp. 10–11, 2018.
8. G. Conte and P. Doherty, "Vision-based unmanned aerial vehicle navigation using geo-referenced information," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, pp. 1–18, 2009.
9. J. P. Golden, "Terrain contour matching (tercom): a cruise missile guidance aid," in *Image processing for missile guidance*, vol. 238. International Society for Optics and Photonics, 1980, pp. 10–18.
10. J. R. Carr and J. S. Sobek, "Digital scene matching area correlator (dsmac)," in *Image Processing For Missile Guidance*, vol. 238. International Society for Optics and Photonics, 1980, pp. 36–41.
11. A. Cesetti, E. Frontoni, A. Mancini, A. Ascani, P. Zingaretti, and S. Longhi, "A visual global positioning system for unmanned aerial vehicles used in photogrammetric applications," *Journal of intelligent & robotic systems*, vol. 61, no. 1-4, pp. 157–168, 2011.
12. C. Grönwall, J. Rydell, M. Tulldahl, E. Zhang, F. Bissmarck, and E. Bilock, "Two imaging systems for positioning and navigation," in *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. IEEE, 2017, pp. 120–125.
13. F. Lindsten, J. Callmer, H. Ohlsson, D. Törnqvist, T. B. Schön, and F. Gustafsson, "Geo-referencing for uav navigation using environmental classification," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1420–1425.
14. M. Mannberg and A. Savvaris, "Landmark fingerprinting and matching for aerial positioning systems," *Journal of Aerospace Information Systems*, vol. 11, no. 3, pp. 131–139, 2014.
15. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

16. M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, and M. Jagersand, "Rtseg: Real-time semantic segmentation comparative study," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1603–1607.

17. D. J. H. Olivier Courtin, *RoboSat.pink Computer Vision framework for GeoSpatial Imagery*, DataPink, 2019. [Online]. Available: http://RoboSat.pink

18. P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, "Learning Aerial Image Segmentation from Online Maps," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6054–6068, 2017.

19. E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark," *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2017-July, pp. 3226–3229, 2017.

Efficient Ego Lane Detection for Various Lane Types

Rebekka Charlotte Peter*, Yuduo Song*, and Martin Lauer

Karlsruhe Institute of Technology
Insitute for Measurement and Control Systems
Engler-Bunte-Ring 21, 76131 Karlsruhe

Abstract In this work, we present an ego lane detector designed for the use in automotive vision systems for personal light electric vehicles like electric bicycles, tricycles or scooters. The approach is based on a combination of gradient-based line detection, color-based segmentation and geometrical rules, making the ego lane detector fast, but also robust to different scenes, including curves. Qualitative evaluation on over fifty traffic scenes show that the lane detector is able to find a suitable approximation of the road area with an IoU of 75.71%.

Keywords Ego lane detection, color-based segmentation, vanishing point detection

1 Introduction

In recent years, personal light electric vehicles like electric scooters, bicycles or tricycles have been gaining in popularity. Being small and lightweight, they represent an emission free alternative to cars or a *last-mile* extension to public transportation systems. To increase safety and comfort of users, automation and driving assistance systems as for autonomous vehicles are conceivable. Even though the use-case appears to be similar, certain differences between personal light electric vehicles and cars make the direct application of algorithms difficult: As the product costs for personal light electric vehicles are significantly lower in comparison to cars, the reasonable

* These authors contributed equally.

maximum costs for sensors as well as computation hardware is lower in the same way. The same applies for a lower possible power consumption of the sensors and computation hardware, as the overall system offers less power. Existing algorithms for autonomous cars must be adapted to new traffic scenes and areas, as personal light electric vehicles are not restricted to drive on streets, but they can also use bicycle lanes or pedestrian paths. Aforementioned differences make especially the application of deep learning methods not readily transferable, firstly, because of the restricted hardware options, secondly, because of the variation in the input data to the training data sets for that the autonomous driving algorithms are optimized for. Moreover, learning-based methods can not merely be retrained because of the lack of datasets including traffic scenes of pedestrian paths and bicycle lanes or related traffic signs etc.

This work presents an algorithm for detecting the two borders of the lane, on which the ego vehicle, more precisely the ego bicycle, is. The above mentioned requirements for low-cost sensors and computation hardware as well as the applicability in various kinds of traffic scenes are fulfilled. Possible applications using ego lane detection include, for instance, obstacle detection on the ego lane or the usage of the ego lane information for traffic scene classification. The lane border detection system works on RGB images taken from a camera mounted on the handle bar of a bicycle. This camera setup and scene perspective is applicable to most kinds of electric vehicles. The lane boundaries are estimated with two straight lines on the left and right side of the lane and, where applicable, a third line at the far end of the visible road area. This approximation is close to the actual ego lane for many cases, but is limited to a straight lane course. In curves, the aim is the linearization of the ego lane at the current position with two straight lines using motion information.

The task entails following challenges: While the borders of streets are often clearly distinguishable from neighboring areas due to lane markings or clear material changes, the transitions can be smoother for pedestrian or bicycle areas, especially where vegetation is adjacent to the lane. Another difficult case occurs if shadows overlap the lane borders or if the lane borders are occluded through dirt or objects such as parking vehicles.

The contribution of this work is the development of a fast ego lane detection system that is suitable for personal light electric vehicle applications as it works in various road places. Using a combination of two line detection strategies and geometrically based rules, no large annotated data set is needed.

2 Related Work

Chougule et al. as well as Meyer et al. present deep learning approaches for lane border detection and lane segmentation in [1] and [2], respectively. Thereby, a mean IoU of 76.39% (cf. [1]) and 80.01% for ego lanes (cf. [2]) is yielded. However, their methods are not suitable for personal light electric vehicle applications with limited computation hardware. Furthermore, due to the dependency on datasets, results are significantly worse for pedestrian or bicycle lanes, as all training samples are taken from a car driver's perspective driving on a street.

Lane detectors based on traditional image processing methods are presented in [3] and [4]. For road area segmentation, in those works texture descriptors are used. We show that a simple distance function based on color information suffice, is faster to calculate and furthermore, better suited for the application on various lane types where the variance in road surface structures is high compared to solely street applications. In [4] and also in [5], the position of the vanishing point is used to enhance lane area prediction. As the geometrical conditions of scenes in driver's perspective give valuable information about the lane borders, we use this approach for selecting the corresponding lines from a set of candidates. We show that a fixed vanishing point estimation is sufficient for the approximation of straight lanes.

3 Methology

The overall system goes through three phases for each image. First, lane border candidates are proposed using gradient and color image information. Secondly, the both candidates who best meet the geometric conditions of the scene are chosen as left and right border

line. Thirdly, based on the movement between the previous and the current image, it is decided whether the ego vehicle is currently driving a curve. If this is the case, with the aid of movement information, straight road lane boundaries are estimated, that linearize the curve at the current position.

Our approach specifically considers the following three traffic scenes:

1. The ego vehicle drives straight on a straight lane. The lane borders are rich in contrast. In this case, the lane borders can be extracted with traditional gradient-based edge detection methods. The approximation of the lane area with straight lines is suitable.

2. The ego vehicle drives straight on a straight lane, but the lane boundaries are not clear due to occlusions (e. g. vehicles parked on the roadside) or smooth transitions (e. g. vegetation at the roadside). In this case, the gradient based approach is unsuitable. Thus, the road surface is extracted using color-based segmentation. The approximation of the lane area with straight lines is suitable.

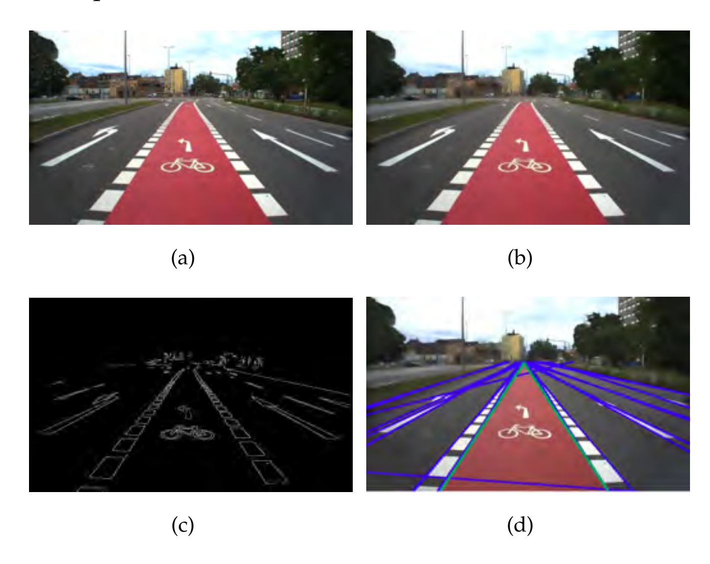
3. The ego vehicle drives along a curve. In this case, the two previous approaches may produce inappropriate results, because the condition of a straight roadway is not fulfilled. The goal in curves is the approximation of the actual roadway by linearization of the lane borders at the current position. For this purpose, the intersection point of the two linearized road edges is estimated using optical flow.

In the following, the three approaches introduced above are described in detail. Then, the final selection of the linearized roadway boundaries is presented. Finally, we quantitatively and qualitatively evaluate the results.

3.1 Gradient-based Lane Border Candidates

If the lane border is rich in contrast, e. g. due to road surface markings or a change in the pavement material, the lane borders can be

extracted with the Canny edge detector pursuant to [6]. To suppress high-frequency noise and high-frequency image structures, a bilateral filter is applied in a pre-processing step. Assuming that the lane borders are dominating lines in the image, they can be found in the gradient image with the Hough line transform according to [7]. The number of proposed lines depend on the scene. Typically, several lines are proposed with the gradient-based approach. See Figure 3.1 for an example.



(a) (b) (c) (d)

Figure 3.1: Visualization of the gradient-based lane border detection. (a) Input image. (b) Bilateral filtering applied. (c) Gradients found with Canny edge detector. (d) Lines found with Hough line transform, including the best candidates for lane border approximation in green.

3.2 Color-based Segmentation Lane Border Candidates

For each image, in addition to candidates based on gradients, color-based segmentation is used to extract the lane area and propose two further candidates using geometrical conditions of traffic scenes. This approach is aimed for situations where the road border line is not clear because of occlusions by plants or other objects. Despite

bilateral filtering, no lines are found at lane borders, if the transition is fluent on the one hand. On the other hand, edges extracted from vegetation does not allow to find the lane border line. Then, a high number of edges in different orientations are found near the actual lane border when the Canny edge detector is applied. For color-based segmentation of the lane area, a region of interest (ROI) is chosen in the lower center of the image. Assuming that most pixels of the ROI show the surface of the ego lane, a binary mask with pixels that may belong to the ego lane is created using a color-based distance function. The reference color is the average of all color values of the pixel in the ROI. Several options of distance functions for color images exist. For our application, best results are archived using a modified version of the CIE94 ΔE^* color distance definition as defined in [8]: For a reference color (L_1^*, C_1^*, H_1^*) and another color (L_2^*, C_2^*, H_2^*) defined in the CIELAB color space, the color distance is defined as

$$\Delta E_{94}^* = \sqrt{\frac{\Delta L^{*2}}{k_L S_L} + \frac{\Delta C^{*2}}{k_C S_C} + \frac{\Delta H^{*2}}{k_H S_H}}, \tag{3.1}$$

with ΔL^* being the lightness difference, ΔC^* being the chroma difference and ΔH^* being the hue difference. S_L, S_C and S_H are weighting functions that adjust the CIE differences $(\Delta L^*, \Delta C^*, \Delta H^*)$ according to the standard in CIE 1976 color space: $S_L = 1$; $S_C = 1 + 0.045C^*$; $S_H = 1 + 0.015C^*$. k_L, k_C and k_H are parametric weighting factors of the three components. To decrease the impact of lightning on the color distance, we choose a high value for k_L. In that way, shadows on the lane surface has less influence on the segmentation result.

By calculating the color distance for each pixel and thresholding, a binary image is created.

For proposing two lane borders in the binary image, the position of the vanishing point is used. We assume a fixed position of the vanishing point for a certain camera setup for simplicity and to show the robustness of our method. An important prerequisite is that the recorded images are in conformity with the perspective principle: Assuming that the left and right lane borders are straight, parallel and run in driving direction, they intersect in the vanishing point

in the image. Thus, assuming a straight and parallel lane, all possible lane border candidates identified from the binary road area image must run through the vanishing point. A second condition is, that the ratio of *lane* pixel to the number of all pixels on the line is higher than a certain threshold. For the color-based line detection, one line for each the left and right lane boundary is proposed that runs through the vanishing point, exceeds the *road pixel threshold* and has the maximum opening angle from all possible lines fulfilling the first and second condition.

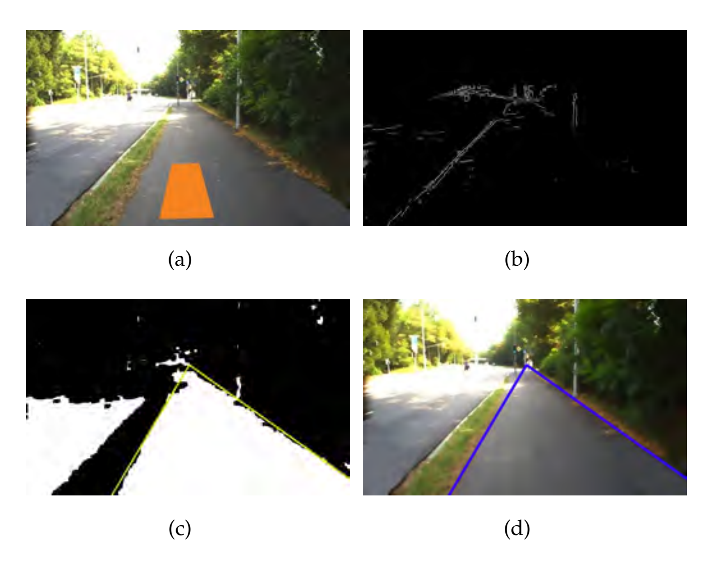


Figure 3.2: Visualization of the segmentation-based lane border detection. As shown in (b), the gradient-based approach fails in this case. (a) Input image with the ROI marked in orange. (b) Edges and lines found with Canny edge detector and Hough line transform. (c) Binary mask: white pixel: color distance to reference color below threshold (*lane* pixel), black pixel: above threshold. (d) Color-based candidates in input image.

3.3 Linearization of Curves using Motion Information

While the two methods above rely only on the current frame, the ego motion between the previous and current frame is used in the cases

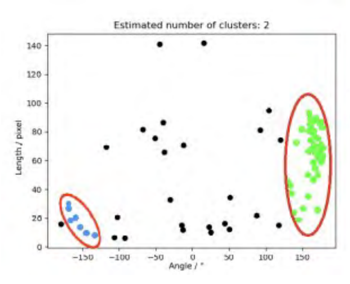
of curves. More precisely, the sparse optical flow between the two frames is used to refine the intersection of the road edges, which was originally set as a fixed vanishing point. The idea is that the projection of the optical flow on the horizontal axis is a measure of how far the intersection point of the two border lines of the road must shift in the direction of the curve in order to achieve a linearization of the road at the current position. The linearization should approximate the actual lane course in the best possible way with straight lines and the IoU between the actual and the linear approximated lane surface should be optimized.

With the Lucas Kanade method, cf. [9], sparse optical flow vectors are calculated for feature points above the estimated horizon in both images. Then, noise, e. g. as a result of mismatched feature points is reduced with two-dimensional Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for the vector length and direction. Details about the clustering method DBSCAN are given in [10]. For vectors of the dominating cluster, the average length in horizontal direction $|\bar{V_u}|$ is determined. To take into account the difference of the distance to the camera between the feature points and the vanishing point at the horizon, the displacement of the original point of intersection (poi), thus, the static vanishing point, is defined as $\Delta_{poi} = \pm|\bar{V_u}|^{1.25}$ with the sign selected according to the vector direction. Figure 3.3 visualizes the optical flow vector, the clustering and displacement of the point of intersection for a sample image.

In curves, the gradient- and color segmentation approach fails as the assumption of straight, parallel lane borders is not fulfilled. Instead, we use the assumption that the scene between two images differs only slightly and take the intersections of the roadway boundaries and the lower image border from the previous image. In that way, three image points are defined, which are the start and end point of the approximated lane boundaries.

3.4 Final Lane Border Selection

In 3.1 and 3.2 it is shown, how several lane border candidates are proposed. Following rules and conditions are applied to find the



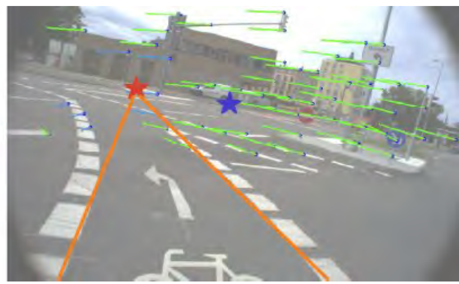


Figure 3.3: Visualization of poi refinement in curves. Left: Optical flow vectors in orientation - length space. The two dominating clusters found with DBSCAN marked with circles. Right: Optical flow vectors of two main clusters in input image. The blue star marks the position of the default vanishing point. The red star is the estimated point of intersection of the left and right lane border (orange lines).

two candidates that represent the left and right lane border most likely:

1. Assuming a straight road, the angle between the road border line and the horizontal image boundary is within a certain range. Experimentally determined are the ranges [30°,80°] and [100°,150°] for the left and right lane boundary, respectively.

2. Assuming straight, parallel lane boundaries, the both lines intersect in the vanishing point. Thus, the condition is set that the absolute horizontal distance of the point of intersection of the lane borders with the horizon to the position of the vanishing point should be below a certain threshold.

3. Of the remaining lines, the two whose intersection with the bottom edge of the image is farthest from the center are selected.

If the mean absolute length of the optical flow vectors $|\bar{V_u}|$ is above a certain threshold, it is assumed that the ego vehicle is driving on a curved lane. Then, two lane borders as described in 3.3 are taken as final selection.

3.5 Evaluation

We evaluated our approach on a total of over 1200 images from about 50 different traffic scenes sequences. The scenes include one- and multi-lane streets, bicycle lanes (separate, distinctly on streets, and besides pedestrian paths), and pedestrian paths, forest paths or parks. In most cases, the lane borders are predicted only with minor deviations from the actual position. Even though the true lane area can not be represented correctly in curves as the lane borders are limited to straight lines, the detected lane area overlaps widely for the majority of test samples. Most scenes for which errors occur, show wide, open roads and a high variations from the standard case of two parallel lane boundaries. For a quantitative analysis, we take the best possible linear approximation with two lines as ground truth. We reach a mean IoU between the area enclosed by the predicted and annotated lane borders and the button line of 75.71%. Figure 3.4 and 3.5 show representative results for straight lanes and curves.

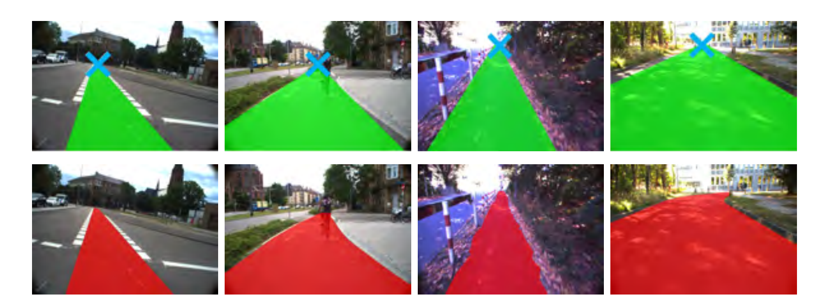


Figure 3.4: Representative results for straight streets, bicycle lanes and sidewalks. Top row: our results. The blue cross marks the point of intersection. Bottom row: ground truth.

4 Discussion and Summary

Although each step of the pipeline is fast and simple, the lane border detector is powerful and yields good results for various traffic types including streets, bicycle and pedestrian lanes comparable to deep

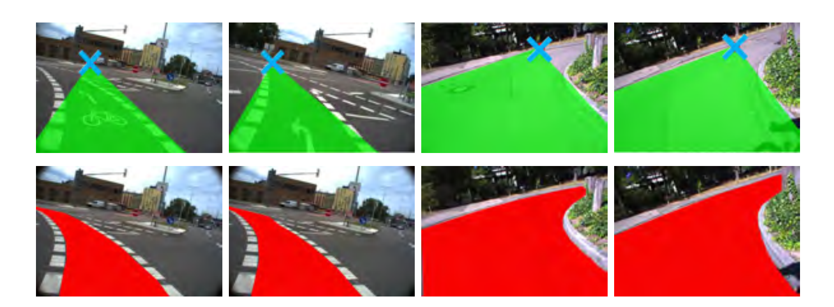


Figure 3.5: Representative results for curved streets, bicycle lanes and sidewalks. Top row: our results. The blue cross marks the point of intersection. Bottom row: ground truth.

learning approaches. Neither a large training data set or ground truth labels are needed, nor are parameters needed to be fine-tuned for the different lane types. Moreover, the algorithm can be to run on low-cost hardware in real-time, which make a great advantage over deep learning based approaches for applications on personal light electric vehicles.

References

1. S. V. Chougule, A. Ismail, G. Adam, V. Narayan, and M. Schulze, "Reliable multilane detection and classificationusing a compact encoder-decoder cnn," in *Forum Bildverarbeitung*, 2018, pp. 291–301.
2. A. Meyer, N. O. Salscheider, P. F. Orzechowski, and C. Stiller, "Deep semantic lane segmentation for mapless driving," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 869–875.
3. S. Graovac and A. Goma, "Detection of road image borders based on texture classification," *International Journal of Advanced Robotic Systems*, vol. 9, no. 6, p. 242, Jan. 2012. [Online]. Available: https://doi.org/10.5772/54359
4. H. Kong, J. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2211–2220, 2010.

5. U. Ozgunalp and S. Kaymak, "Lane detection by estimating and using restricted search space in hough domain," *Procedia Computer Science*, vol. 120, pp. 148–155, 2017. [Online]. Available: https://doi.org/10.1016/j.procs.2017.11.222
6. B. Jähne, *Digitale Bildverarbeitung*. Springer Berlin Heidelberg, 2002. [Online]. Available: https://doi.org/10.1007/978-3-662-06731-4
7. J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
8. R. McDonald and K. J. Smith, "Cie94-a new colour-difference formula*," *Journal of The Society of Dyers and Colourists*, vol. 111, pp. 376–379, 2008.
9. B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (ijcai)," vol. 81, 04 1981.
10. "A density based algorithm for discovering density varied clusters in large spatial databases," *International Journal of Computer Applications*, vol. 3, 06 2010.





Bildverarbeitung spielt in vielen Bereichen der Technik zur schnellen und berührungslosen Datenerfassung eine Schlüsselrolle. Beispielsweise in der Qualitätssicherung industrieller Produktionsprozesse, in der Robotik und zur Fahrerassistenz haben sich Bildverarbeitungssysteme einen unverzichtbaren Platz erobert. Diese Entwicklung wird unterstützt durch die Verfügbarkeit qualitativ hochwertiger und günstiger Sensorsysteme sowie durch die Zunahme der Leistungsfähigkeit von Rechnersystemen.

Der vorliegende Tagungsband des „Forums Bildverarbeitung", das am 26. und 27. November 2020 in Karlsruhe als gemeinsame Veranstaltung des Karlsruher Instituts für Technologie und des Fraunhofer-Instituts für Optronik, Systemtechnik und Bildauswertung stattfand, enthält die schriftlichen Aufsätze der eingegangenen Beiträge. Darin wird über aktuelle Trends und Lösungen der Bildverarbeitung in den methodischen Schwerpunkten Bildgewinnung, 3D-Verfahren, Bildverarbeitung, Maschinelles Lernen und Navigation berichtet.

ISBN 978-3-7315-1053-6


Gedruckt auf FSC-zertifiziertem Papier